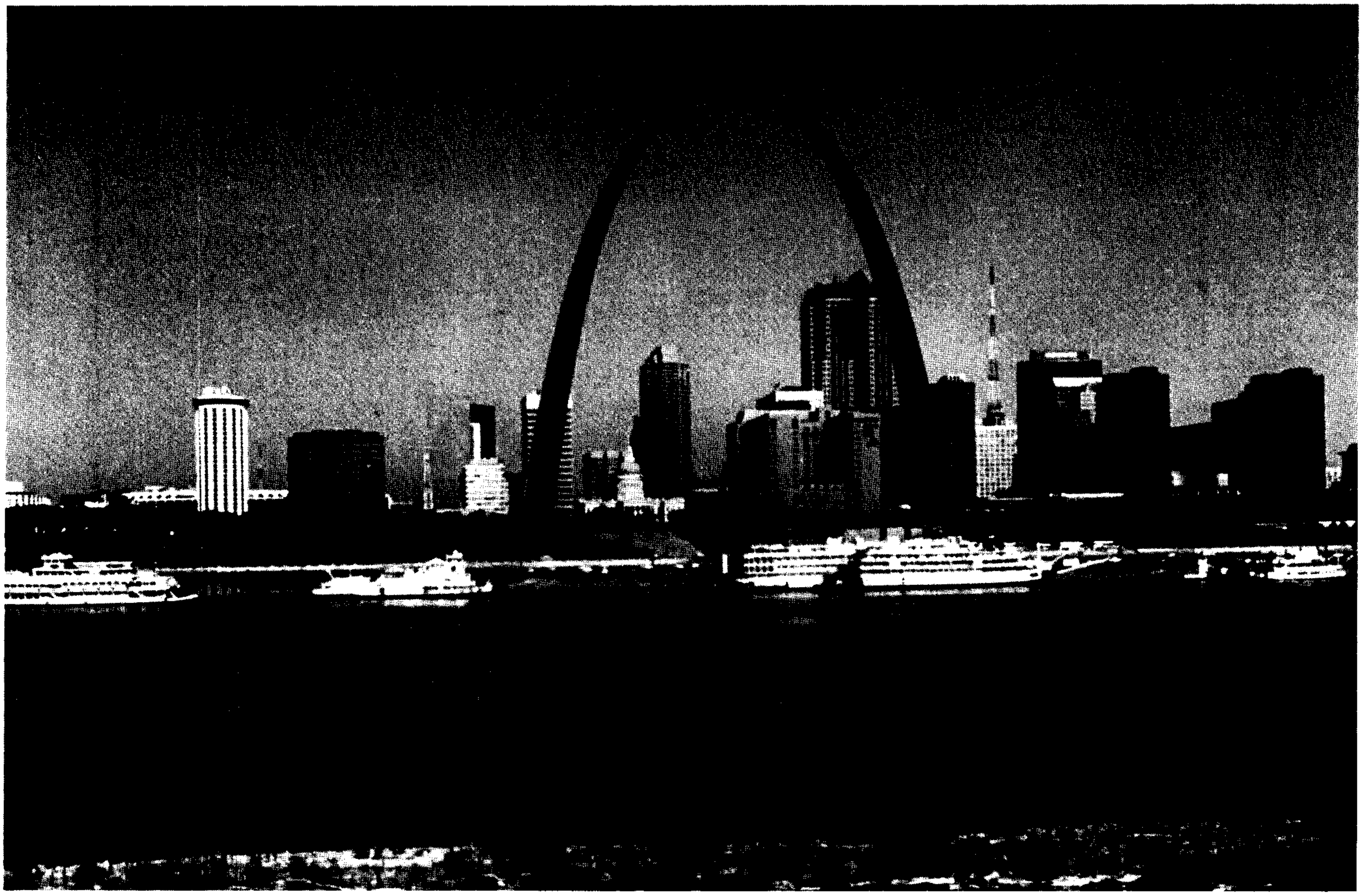
ISECON '96



Gateway to the Future: Proceedings of the 13th Information Systems Conference

St. Louis , Missouri
October 18 - 20, 1996

Sponsored by the DPMA
Education Foundation

PROCEEDINGS

ISECON '96

INFORMATION SYSTEMS EDUCATION CONFERENCE

GATEWAY TO THE FUTURE

St. Louis, Missouri
October 18 - 20, 1996

Joyce Currie Little, Editor
Towson State University

Data Processing Management Association
Education Foundation
505 Busse Highway, Park Ridge, IL 60068-3191
(847) 825-8124
http://www.dpma-ef.org
Dedicated to engancing educational opportunities for information systems professionals

Dear Information Systems Educators

On behalf of the DPMA Education Foundation Board of Regents, it is with great pleasure that I welcome you to ISECON '96: Gateway to the Future, in St. Louis, Missouri. This is an excellent opportunity to learn from and share ideas with your peers regarding the future of Information Systems Technology.

The Conference Committee, under the chairmanship of Dr. Bruce White from Dakota State University, has put together a great conference. The educational programs continue to be the best offered anywhere and we hope that you will take full advantage of them.

While you are in St. Louis where the Gateway Arch is a symbol of spirit and achievement, we hope that you can take advantage of some of the attractions for visitors of this great city.

Be sure to attend the Sunday luncheon at which we recognize the DISEA winner. This award, sponsored jointly by the DPMA Education Foundation and DPMA EDSIG, annually recognizes the educator of the year. This event is a highlight of the conference, since the award recipient always has an interesting presentation and allows all of us to pay tribute to an outstanding individual. All of the elements of a fantastic educational experience is here for you.

Welcome to ISECON '96: Gateway to the Future.

Best Wishes,

Lou

Louis J. Berzai, CSP
President
DPMA Education Foundation
Board of Regents

DAKOTA STATE UNIVERSITY
Madison, South Dakota 57042-1799

WELCOME TO ISECON '96

Information Systems Education: The Gateway to the Future

I am pleased to welcome you to the 1996 ISECON conference. On behalf of the Board of Regents of the Education Foundation of DPMA, the steering committee and EDSIG, I would like to thank you for your attendance and participation in ISECON '96.

In this year's conference, you will find some exceptional presentations on Information Systems Education. As with past (and future) ISECON conferences, the toughest decisions may be which of the sessions to attend and which to read about in the proceedings. It's time to make and renew acquaintances in the field; to check out the latest offerings from the exhibitors; and to be invigorated by speakers, panels and workshops on Information Systems Education topics.

Be sure to catch the keynote address on "Education to Solve Software's Chronic Crisis. Take in one of the special practical workshops - from curriculum design to object oriented COBOL and topics in-between. Sunday's luncheon is always a highlight of the conference as the EDSIG / Education Foundation's Distinguished Information Systems Educator of the year Award is presented, and the recipient tells of her research in the field. Take time to reflect, and to bring your ideas and successes back next year as a presentation to ISECON '97 to be held in Orlando.

Sincerely,

Bruce White

Bruce A. White, Ph.D.
Conference Chair

ISECON '96
GATEWAY TO THE FUTURE

LEADERSHIP TEAM

CONFERENCE CHAIR
Dr. Bruce White
Dakota State University
Madison, South Dakota

DPMA-EDUCATION FOUNDATION LIAISON
Dr. Lou Berzai
Notre Dame University
South Bend, Indiana

PAST CHAIR
Dr. Eli Cohen
Wollongong University
Wollongong, Australia

PROCEEDINGS
Dr. Joyce Currie Little
Towson State University
Baltimore, Maryland

PROGRAM CHAIR
Dr. Roy Daigle
University of South Alabama
Mobile, Alabama

PANEL/WORKSHOP CHAIR
Dr. Al Harris
Appalachian State University
Boone, North Carolina

ISECON '96
GATEWAY TO THE FUTURE

THE PROGRAM COMMITTEE

REFEREED PAPERS

Dr. Roy Daigle
University of South Alabama
Mobile, Alabama

PANELS AND WORKSHOPS

Dr. Al Harris
Appalachian State University
Boone, North Carolina

TRACK CHAIRS

Student Track:
Mark Hensel, University of Texas at Arlington

Small Colleges, Two-Year or Community College Programs:
Pam Chapman, Waubonsee Community College

Emerging and Leading Edge Technologies:
Eugene McGuire, American University

Industry and Information System Education Relationships:
Jim Pick, University of Redlands

International Information System Issues:
Anton J. van Reeken, University of Limburg

Curriculum:
Michael Doran, University of South Alabama

CIS Program Directions and Issues:
Marino Niccolai, University of South Alabama

ISECON '96
GATEWAY TO THE FUTURE

REFEREES

Elizabeth Adams, Richard Stockton College of New Jersey
Yvonne Lederer Antonucci, Widener University
Pam Chapman, Waubonsee Community College
Carol Chrisman, Illinois State University
Mike Doran, University of South Alabama
Barbara Ehhard, Northern Illinois University
Irving Englander, Bentley College
Maureen Foley, American University
Wanda Grabow, College of DuPage
Ruth Guthrie, University of Redlands
Susan Helms, Metropolitan State College of Denver
Catherine Blake Jaktman, University of Technology, Sydney, Australia
Janet Kemp, University of South Alabama
Marino Niccolai, University of South Alabama
Patrick Olson, University of redlands
Bridget O'Connor, New York University
William Owen, University of South Alabama
Joan Ozello, George Mason University
J. Harold Pardue, Florida state University
Dave Petrie, University of Redlands
William P. Qualls, Waubonsee Community College
Mary Ann Robbert, Bentley College
Donna Schaeffer, University of Redlands
Karen Schenk, North Carolina State University
Anton J. van Reeken, University of Limburg
Sharon Vest, University of South Alabama
Les Waquespack, Bentley College

ISECON '96
PROCEEDINGS

TABLE OF CONTENTS

FRIDAY, OCTOBER 18

SATURDAY, OCTOBER 19

WORKSHOP

OBJECT-ORIENTED COBOL

Edmund C. Arranga
The COBOL Group

Object technology is taking the software world by storm. Objects are the future; by the year 2000, 80% of all distributed computing will be object-oriented. Already objects have been responsible for a metamorphisis across all phases of the software lifecycle, spawning new subdisciplines that include object-oriented analysis and design, object-oriented databases, object-oriented operating systems, and object request brokers.

Objects are not limited to a single platform; they are equally at home on a personal computer (PC) as on a mainframe, client-server network, or the global Internet. Objects are not a fad but the extension of abstraction and encapsulation techniques that have been evolving since the earliest versions of COBOL.

Object-Oriented COBOL is the latest evolutionary step in COBOL's history. COBOl is a survivor with a long, rich, colorful, and at times turbulent history of adaptation to change. Although COBOL's robustness and scalability have served it well, the limitations of the structured paradigm have taken their toll. Today COBOL stands at the crossroads of two paradigms: structured methods and object technology. Procedural COBOL and structured methods have outgrown their own rules of self-regulation and anecdotal techniques. It is time to move ot the futre. It is time to move to Object-Oriented COBOL.

Come join with us in a day long seminar as Edmund C. Arranga explains Object-Oriented COBOL.

Session 1: Learn the syntax and semantics of Object-Oriented COBOL, including classes, ojbects, and message passing.

Session 2: Understand how inheritance, polymorphism, and abstract classes are implemented in Object-Oriented COBOL.

Session 3: Participate in a small object-oriented analysis and design session and learn why progrmming with objects is not necessarily object-oriented programming.

An Introduction to Developing Multi-Tier Client-Server Applications Using PowerBuilder® 4.0

Mayur R. Mehta, Southwest Texas State University, San Marcos, Texas 78666
George W. Morgan, Southwest Texas State University, San Marcos, Texas 78666

Introduction:

Every decade the field of computer information systems seems to undergo major changes. 1970s saw the birth of on-line transaction processing systems and management information systems. In the Eighties, the focus shifted towards PCs, executive information systems and expert systems. The 1990s have already given us powerful microcomputers, GUI, OOP, powerful database servers, and of course, client-server architecture.

Downsizing, rightsizing and client/server computing have become the hottest topics in the IS field. With this shift towards the client/server paradigm has come a sleuth of new application development tools. These visual programming tools enable IS professionals to rapidly develop industrial-strength client-server applications that permit users to retrieve and manipulate enterprise-wide information seamlessly and effortlessly using a GUI front-end.

In addition to the client/server paradigm shift, the emerging technology has also brought about another major problem. Industry IS managers have found it difficult to recruit people who are skilled in the use of newer application development tools. Universities across the nation are revising their curriculum to meet the changing demand and expose students to the new breed of application development tools.

In recent years, PowerBuilder® has become a dominant client-server application development tool. PowerBuilder® provides cross-platform, full-function GUI features coupled with the ability to easily connect to a wide variety of database servers. For this reason, it has become the application development tool of choice for developing applications that are flexible, scaleable, and that can exist on a multi-platform network.

This workshop is designed to introduce the novice participant to the use of PowerBuilder® in developing robust client-server applications. The workshop will strive to achieve the following objectives:

Objectives of the Workshop:

1. Provide participants new to the client-server world an understanding of the concepts behind the client-server paradigm and multi-tiered client-server applications development methodology.
2. Present an understanding of event-driven, GUI programming techniques,
3. Introduce the participant to the rich application development features offered by PowerBuilder®,
4. Demonstrate the use of PowerBuilder® in developing multi-tiered client-server applications.

Workshop Outline:

1. The Client-Server Paradigm
    - 1.1. What is Client-Server Computing
    - 1.2. Approaches to Implementing Client-Server Paradigm
    - 1.3. Functional Application Distribution
        - 1.3.1. Presentation Logic
        - 1.3.2. Application Logic (Business Rules) Information Retrieval/Management Logic
2. Event-Driven, GUI Programming Techniques
3. Introduction to PowerBuilder® Environment
    - 3.1. PowerBuilder® Painters
    - 3.2. Building Graphical User Interface Front-End
        - 3.2.1. Single Document Interface (SDI) vs. Multiple Document Interface (MDI) applications
        - 3.2.2. GUI Controls/Objects
        - 3.2.3. GUI Design Paradigm
4. Building Back-End Database Server
    - 4.1.1. Using Watcom
    - 4.1.2. Using MS-Access
5. Connecting GUI Front-End to Database Server via PowerBuilder Datawindows
    - 5.1.1. Constructing SQL
    - 5.1.2. Selecting Presentation Format
    - 5.1.3. Need for Middleware - ODBC drivers

A short case study will introduce participants to a step-by-step approach to developing client-server applications using PowerBuilder®.

Workshop
Introduction to Visual Basic
Cathy Bishop-Clark

This 90 minute workshop will introduce educators to the main features of programming in the event-driven visual programming language—Visual Basic. Select components of the Visual Basic environment will be illustrated. Although the workshop will not be hands-on, we will as a group interactively develop some simple Visual Basic Applications.

Visual Basic allows programmers and novice students to create highly visual, relatively sophisticated programs with little experience. It is an exciting and engaging language. Although introduced only recently it is becoming quite popular. The syntax of Visual Basic is easy and the environment support quick implementation. This gives a beginning student a sense of achievement and a feeling of having control. Until recently, most of the books available for Visual Basic were reference type; however, several introductory textbooks appropriate for an introductory programming class are currently available. Just as C/C++ has replaced Pascal in many universities, it may be just a matter of time before Visual Basic will replace other versions of Basic in introductory courses.

A outline of the topics is shown below:

I. History/Overview of Visual Basic

II. Overview of the Visual Basic Environment
- a. Form
- b. Properties
- c. Tools
- d. Project Box
- e. Help

III. Designing the User Interface—select tools
- a. Labels
- b. Picture Boxes & Images
- c. Text Boxes
- d. Commands
- e. Frames
- f. Combo Boxes & List Boxes

IV. Programming Structures
- a. Data Types
- b. Basic Control Structures

V. Simple Example Programs

VI. Visual Basic Resources

PANEL SESSION

DESIGNING A CIS CURRICULUM TO MEET INDUSTRY SPECIFICATIONS: A PROJECT OF THE NATIONAL SCIENCE FOUNDATION

PANELISTS:

David L. Feinstein
School of Computer & Information Sciences
University of South Alabama

Doris K. Lidtke
Towson State University

Michael C. Mulder
National Science Foundation

The National Science Foundation is funding a three-year effort to define a curriculum and a delivery mechanism for Computer Information Science. The unpreecedented growth of Computer Information Science and the need for qualified practitioners in the field have motivated this curriculum design. One of the unique aspects of this project comes from the composition of th task force. Half of the group comes from business, industyr and government, and the other half from academia. Representatives from aerospace, banking and telecommunication are actively involved in the definition of the model.

The industry representatives have developed a Profile of the Graduate specifying th ideal characteristics for a new hire. The entire task force is developing the curriculum to meet the industry specifications. The panel will present the Profile of the Graduate and the resulting first year curriculum. Initial pilot sites have been chosen and initial implementation has begun. This has included a strong component of joint student and industry professional teams working on mission critical applications.

Building Cool Worlds With VRML

Lynn R. Heinrichs, Western Carolina University

Introduction

Faculty experienced in developing HTML documents may be ready for the next step in World Wide Web publishing: building 3-dimensional, interactive environments. One tool available for creating such "cool worlds" is the Virtual Reality Modeling Language (VRML). VRML worlds can be embedded in HTML documents to breathe life into World Wide Web pages. WWW authors can use VRML to rotate objects, move through rooms and buildings, and fly over landscapes to name only a few applications.

Many of the same questions surrounding HTML development also surround using VRML. How does one create a cool world? Is the technology complex to use? Does the technology provide an effective vehicle for communication? Will performance be acceptable to users? What type of resources are required? How much does it cost to get started? The purpose of this workshop will be to provide an overview of VRML for faculty who are interested in (1) developing their own worlds or (2) providing VRML coverage in the information systems curriculum.

Workshop Outline

I. Introduction to VRML
   A. History
   B. Applications
   C. Terms and Concepts

II. Building Cool Worlds
   A. Coding by hand
   B. Using an authoring package
   C. Demonstration
   D. Design considerations

III. Resource requirements
   A. Software
   B. Hardware
   C. Cost

IV. Role in the IS curriculum
   A. World Wide Web/Internet
   B. Multimedia
   C. Application development

V. Getting Started

VRML World Wide Web Resources

Cool Worlds
www.netscape.com/comprod/products/navigator/live3d/cool_worlds.html

Introduction to VRML
www.netscape.com/comprod/products/navigator/live3d/intro_vrml.html

VRML Repository: Software/Modelers
www.sdsc.edu/SDSC/Partners/vrml/software/modelers.html

VRML Repository: Documentation
www.sdsc.edu/SDSC/Partners/vrml/doc.html

Virtual Reality Modelling Language (VRML) Forum
vrml.wired.com

VRML Repository
sdsc.edu/vrml

VRML
www.vrml.org

VRML Update
cedar.cic.net/~rtilmann/mm/vrmlup.htm

NCSA VRML Home Page
www.ncsa.uiuc.edu/General/VRML/VRMLHome.html

webdog's Virtual Reality
www.webdog.com/vr.html

ExpressVR Home Page
www.cis.upenn.edu/~brada/VRML/ExpressVR.html

Virtual Reality & VRML Library
cedar.cic.net/~rtilmann/mm/vrml.htm

Yahoo - Computers and Internet:Internet:World Wide Web:Virtual Reality Modeling Language (VRML)
www.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/Virtual_Reality_Modeling_Language__VRML_/

Web Developer's Virtual Library: VR
www.stars.com/Vlib/Providers/VR.html

KEYNOTE ADDRESS
EDUCATION TO SOLVE SOFTWARE'S CHRONIC CRISIS

Dr. Carl Clavadetscher, National Defense University
(202)685-4881
clavadetscherc@ndu.edu

<u>Introduction</u>: In IS education we can derive our IS curricular standards from <u>internal</u> or <u>external</u> sources. If we choose <u>internal</u>, we will surely become a self-propelled auger as we drill and drill until we disappear from sight into a hole we have created.

<u>Good news</u>: demand is high for our graduates; system development has never been more central to national success. IS is at a state of maturity where we get attention, often negative.

<u>Bad news</u>: system development efforts, within the private sector, government agencies, and DoD, are notoriously unsuccessful, a batting average below .100. We are under attack for that.

<u>Causes of the failures</u> are institutional and process-control related. To write code/requirements is to make "misteaks." Catching mistakes is the key. The SEI CMM model labels most software development efforts as "level I," or "individuals and heroics:" no foundation software process control structures, individuals fighting battles.

<u>The Key</u>: To the extent we don't teach those software development practices necessary to move above and beyond level I of the CMM, we educate for "individual and heroics" behavior and we are, sadly, a part of the problem.

Not only is industry under attack, IS/CS education is also. "Software Engineering" advocates are pushing for licensing of software engineering professionals, graduates from "E" schools. SE advocates are accurate in their criticisms of the ACM-CS curriculum. Will we respond to them?

Our curricular efforts currently are programmer centered. Can we live with a shift to "software system development" centered? Can we de-emphasize development of code in many syntax's, to emphasize processes which are critical to development of industrial strength software systems?

Can we co-opt the attacks of the SE folks and incorporate industrial best practices for software systems development in our curriculum? The UMUC/UMCP "Master's in Software Engineering" program, taught outside of the engineering disciplines, focuses on building robust software. Can we use this at the undergraduate level? I believe that we must borrow from it.

Can we incorporate more industrial connection in our curricular process, as with the NSF effort? Within DoD we have "functional boards" of qualified professional who develop competencies for information resource management and software acquisition, which then drive the instructional design and delivery process. It works.

Our graduate should be able to produce industrial strength software in a commercial environment. I pose an Information Systems Education Maturity Model: (1) The '95 DPMA Curriculum, (2) industrial practice, (3) and "best practices" coursework.

(Dr. Clavadetscher teaches software project management courses in DoD.)

Crafting A Business Bachelor of Science Program in Information Systems Following on IS'95

Les Waguespack, Donald Chand, Doug Robertson

Computer Information Systems Department
Bentley College, Waltham MA 02254
lwaguespack@Bentley.edu
(617) 891-2908

ABSTRACT

IS'95 has been long awaited to give direction and scope to the information systems education field. Although expansive in its breadth it does not prescribe a framework for implementation and IT faculty will be challenged to do it justice. This paper is a brief description of one attempt to frame the scope and content of IS'95 in an AACSB accredited bachelor of science program consisting of only 30 major credit hours. We present the structure and content of the BSCIS'96 program, briefly review the pressures that led to our revision effort, and discuss a pedagogical innovation in our design that facilitates our framework.

Introduction

For the last three years an interdisciplinary and inter-industrial panel of curriculum architects have been formulating a vision for preparation of IS professionals via higher education programs. The result of their labors is called IS'95, *Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems* [1]. This model catalogs an expansive range of computer science, information systems, management and organizational behavior skills, technologies, methodologies, and professional practices that form the scaffolding of the IS profession. As valuable as is the contribution of this model curriculum in its effective enumeration of computing issues, it also presents a bewildering challenge for academic program designers. Designers must fashion a current day degree program that is deliverable, marketable, "staff-able," "equip-able," affordable and can fill an accredited slot in an AACSB school of business.

Bentley College in Waltham, Massachusetts is a private business college enrolling approximately 3,900 undergraduate and 2,000 graduate students matriculating in all the classic disciplines of business. The Computer Information Systems department has offered a BSCIS degree for nearly 15 years. The BSCIS program has enjoyed national visibility and has graduated some 2,500 students. All of the business programs at Bentley are AACSB accredited.

Following the national pattern of IS program enrollments the CIS student body grew to nearly 500 in the mid-80's and drew down to about 100 two or three years ago. Currently around 125, the BSCIS program is again growing.

Our Legacy BSCIS Program

Within the AACSB framework, BS students are required to complete at least as many course credit hours in non-business coursework as in business coursework. Bentley has chosen to implement the business core body of knowledge, CBK, as a ten, three credit hour course suite of business courses offered by the respective discipline departments. Similarly, the CIS major requirement consists of ten, three credit hour courses. This included seven required courses and three elective choices among CIS courses. The legacy program is summarized below in the table.

Before BSCIS'96
(each 3.0 credit hours)

Required	
CS145	Foundations of Programming
CS245	Program, Data and File Structures
CS201	Structured Programming in COBOL
CS312	Systems Analysis and Design I
CS340	Data Communications
CS350	Data Management
CS395	Systems Analysis and Design II
Electives	
CS295	Computer Graphics
CS300	Security, Accuracy, and Privacy
CS335	Software Development using C and UNIX
CS380	End-User Computing
CS390	Software Engineering
CS421	Internship

Legacy Program Pressures

A variety of market, academic, and institutional pressures developed in the last several years that precipitated the curriculum revision that is reported here. The market for our graduates has become increasingly sensitive to specific technology education and skills with those technologies. This was conveyed directly to the faculty via recruiters and indirectly via student surveys. They requested more technology based electives and expressed more interest in the technology being employed in a particular course than the concept or theory focus of the course. In a parallel, almost symmetric pattern, college administrators and the board of trustees expressed concerns that our program might not be "current." In many cases this could be traced to the retention of COBOL and a lack of "hot button" development tools or languages such as C++, Visual Basic, or PowerBuilder™. This perception existed despite the fact that these tools and languages have been introduced, often demonstrated, and sometimes made available for course related individual and group projects.

Academic pressure was felt in the people costs of maintaining course currency in the face of fast evolving technology. Courses such as programming, systems analysis/design and data management were faced virtually every semester with good reasons for incorporating a new software version, a new tool, or even a new technology into a course which primarily focused on teaching the concepts and principles underlying professional practice. With the number of majors near 120, no more than three sections of such a course could be offered per year. The preparation cost associated with integrating a new technology (e.g. configuration, faculty training, student exercise development, lab resources, project administration) routinely discouraged early adoption of new technology. The preparation cost often prohibited any adoption since it was folded into the "faculty course load" of the 3.0 hour course.

Institutional pressure is that commonly found in any accredited higher education enterprise these days: higher production (class size, teaching load, scholarship) and cost controls (limited lab support in equipment or personnel, limited course development funding in grants or summer stipends). In addition, competition for "shelf space" in the semester course schedule limits the number of offerings that can be made available for student choice.

All these pressures (and others too many to present here) were named and described via a series of faculty task studies on the legacy BSCIS. By examining each of the CIS course offerings and the business curriculum context within which the department must thrive, the resulting findings combined to move the CIS faculty to revise the BSCIS curriculum. The revision addresses these pressures and it does so cognizant of the national guidelines proffered in IS'95.

BSCIS'96 Innovating Through Structure

A central recurring theme in the BSCIS'95 review was the collocation of pedagogy for the concepts or theories of computing with specific technologies related to the professional practice of those concepts. On the surface this seems logical and even compelling since this form has existed in our program (and in most others) for the last 15 to 20 years. Over those years, however, there has been a dramatic shift in the speed at which the technology components evolve. On the one hand the body of concepts and theory reached a degree of maturity that may be considered generally stable. On the other hand the pace of new technology development and introduction renders virtually all IS technology changing (unstable). As a result, course development had become somewhat "schizophrenic." The pedagogy for the theory and concepts cries out for carefully crafted abstractions that lead to effective generalizations of theory and principle. The pedagogy for technology cries out for more tools, more manuals, more hands on, and more implementation detail.

We hypothesized that separating the two paradigms of pedagogy might offer benefits. We then set out to identify those areas of pedagogy that might be so partitioned. We settled on two core areas of the IS'95 framework: the teaching of development through programming and the teaching of modeling through analysis, design and data modeling. We assembled the current syllabi of courses for these two areas and proceeded to restructure them. We used a model of three credit hour concept course "orbited" by appropriate one and one half credit hour technology "satellite(s)."

The resulting model of pedagogy is based on the notion that the mature and more stable concepts and theory are best presented in the three credit hour lecture / seminar format where the focus is on abstraction and the development of patterns of thinking about problems. The complementary notion is the professional technology component that supposes a co- or prerequisite of the relevant concept course, but focuses on the behavioral details of the technology, tool or language. And it provides a hands-on laboratory experience that is dependent more on coaching or mentoring in professional practice rather than upon lecturing in the classroom.

Concept / Technology Component Structure Impacts

The restructured pedagogy presented immediate prospects for combating the market, academic, and institutional pressures we had identified. For the markets it is now possible to designate the specific technology (e.g. Visual Basic™, PowerBuilder™, SQL, Smalltalk™, etc.) directly in the course title listed in the catalog, the course schedule and (most importantly) in the students' transcripts. Electives of a variety of technologies (e.g.

Delphi™, Filemaker Pro™, Visible Analyst™, COBOL, C++, etc.) may be offered as introductory or advanced technology components (1.5 hours credit) without the major ramping up required of the concept courses and without consuming a full three credit hour course slot in either the student's schedule or a faculty member's teaching load.

The specific technology can be replaced without impacting the syllabus of the prerequisite concept course and the concept course can pick and choose among available technologies for in class demonstration without committing to full dissemination to each student or lab station. Both the pedagogy of the concept course and that of the technology component course benefit from the "undivided" attention the course material enjoys from both teacher and student.

The faculty who teach the concept course are not automatically committed to teaching the technology component. Effective teachers of concepts and theory may be interchangeable among concept courses. This becomes practical when movement from one concept course to another does not presume adopting a new tool or language simultaneously. Faculty interested in particular technologies but unable to justify a full concept course, may wish to offer a technology component as a general elective for student breadth experience or as an advanced elective extending a previous introductory treatment in that technology. *[Faculty interest initiated electives may be particularly useful for offering technology to non-majors. Introductory technology components offer students an opportunity to enrich their resumés for job search and sample computer technology potentially leading to a CIS minor or major.]*

Institutionally, the new structure allows technology education to be housed in a full-fledged academic course with catalog description, specific teaching load, schedule, enrollment, teaching evaluation, and appropriate course development support. This uncovers the role that technology plays in our curriculum and the cost that that technology exacts. *[We in CIS have noticed "load creep" where the technology components of our course preparation are either invisible or ignored in the assessment of faculty "production."]* Some of the technology components can or should be delivered by adjunct faculty who in the practice of their primary profession are better versed in the details of a particular technology than a faculty member could normally be. At the same time, such education would enjoy the co- or prerequisite concept coursework necessary to maintain the appropriate academic level and rigor often lacking in activities labeled as "technology training."

Summary of the Restructuring Effort

The course restructuring that is described in this paper did not require changes in either the general education or the business CBK college requirements. Those aspects evolve independently of the major requirements. The CIS changes are totally compartmentalized within the thirty credit hours from which the new program was hewn.

We have focused on the concept / technology component facet of the new BSCIS'96 revision in hopes of stirring a further exchange of ideas. We have tried to use this limited space to share our experience in attempting to benefit from IS'95 in our curriculum development. IS'95 shows the breadth and depth of the IS discipline. IS will continue to be a challenge to colleges and universities.

Special thanks to the collaborative spirit and teamwork of the Bentley College CIS faculty: Anderson, Chand, Drake, Englander, Gorgone, Green, Grillo, Kallman, Linderman, Robbert, Robertson, Saraswat, Waguespack and Zbyszynski.

[1] *IS'95, Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems*, Couger, J. Daniel, Davis, Gordon G., Gorgone, John T., Feinstein, David L., and Longenecker, Herbert E. Jr., School of CIS, University of South Alabama, Mobile, AL 36688, Fall'95.

APPENDIX

Catalog Description: Computer Information Systems

Effective use of information technology will be the mark of successful enterprise in the 21st century. Virtually every business policy decision is intertwined with the application of information technology to plan, produce, market, manage, and service business functions. Careers based on a broad understanding of information technology and expertise in the analysis, design and development of information systems will continue to grow in number and importance well beyond the year 2020. The Computer Information Systems bachelor's degree prepares graduates for just this promising professional future.

As a business major the BSCIS student learns the context and practice of business through courses in the business common body of knowledge. This understanding prepares the BSCIS graduate to be a full business partner who is aware of the interrelationships and inter dependencies in and around modern enterprise in the commercial, government, and not-for-profit sectors. And, students are taught to recognize ethical implications in all business situations and how to incorporate these in business decision making. Upon this business foundation the BSCIS program equips the graduate with not only the knowledge and understanding of information technology capabilities and implications, but also the expertise in "best-of-breed" methodologies and tools for information systems development. The BSCIS graduate is prepared to excel in any of the emerging and varied roles of IT professional: business analyst, systems analyst, application developer, systems integrator, IT liaison, end-user support, network manager, vendor representative, and technical support specialist.

The BSCIS student learns through concept focused and applied technology focused courses. The three credit hour concept courses emphasize the theory and principles of information technology, computer science and management that form the discipline of information systems. The applied technology courses are one and one half credit hours. These emphasize a leading edge technology used to build information system solutions. These courses are designed to develop self-confidence, expertise and a "can-do" attitude in BSCIS graduates. These lab based, applied technology courses allow a close working relationship between student and CIS instructor.

Top students in the program are highly encouraged to include an honors internship program in their course sequence to provide an on-the-job experience prior to graduation.

Required Computer Information Systems Major Courses (30.0 total credit hours)
{CS101 is required general education for all students}

(24.0 credit hours of CIS core:)
CS210 Business Through Information Technology (3.0 credits)
CS220 Computer and Information Structures (3.0 credits)
CS230 Algorithm and Data Abstractions (3.0 credits)
CS231 Programming Environment - Visual Basic (1.5 credits)
CS232 Application Development - Visual Basic (1.5 credits)
CS340 Computer Networks I (3.0 credits)
CS360 Analysis, Modeling and Design (3.0 credits)
CS361 Data Management with SQL (1.5 credits)
CS362 Computer Aided Systems Engineering (1.5 credits)
CS460 Applied Software Project Management (3.0 credits)

(3.0 credit hours of CIS elective)
Students must choose one of the following electives:
CS401 Thesis/Directed Study in Computer Systems (3.0 credits)
CS402 Special Topics [Variable Title] (3.0 credits)
CS421 Internship (3.0 credits)
CS440 Computer Networks II (3.0 credits)
CS450 Object Oriented Technology (3.0 credits)

(3.0 credit hours of CIS or IT electives)
Students may choose one or more of the following as well as any of the remaining four electives above:
Each of the following will be offered with a specific technology in its course title and is one and one half hours credit.

CS212 General Education IT Component
CS221 Operating System Technology I
CS233 Application Programming I
CS313 Intermediate General Education IT Component
CS322 Operating System Technology II
CS334 Application Programming II
CS363 Advanced Data Management Technology
CS364 Rapid Application Development Technology
CS414 Advanced General Education IT Component
CS451 Object Oriented Programming Environment
CS452 Object Oriented Application Development

Computer Information Systems

CS101 Information Systems from an Individual Perspective (three credits)

Provides a comprehensive and current introduction to information processing and the Bentley computing environment. Focuses on understanding the purpose and underlying concepts of information technology and its place in problem solving. The foundation of the course is the Information Processing Model as it relates to individual computing needs. Various operating environment are explored. In addition, the technological, ethical and social dimensions of emerging information technologies are investigated.

CS210 Business Through Information Technology (three credits)
Prerequisite: CS101

This course describes the role of information technology in the conduct of contemporary business. It focuses on the use of IT to facilitate business change in policy and practice. And it discusses the pressure that IT itself has on changes in business policy and practice. It examines IT in retail, production, financial and managerial business activities. This course introduces the students to the varied roles that IT plays in business and serves as a foundation for business case study of particular technologies.

CS212 General Education IT Component (one and a half credits)
Prerequisite: CS210

This course teaches a contemporary IT technology by using a computer based software package. Students are expected to perform operational exercise to gain experience and facility with the particular technology designated for this course section. Students have a broad choice of technology appropriate for students with little or no technical experience beyond CS101.

CS220 Computer and Information Structures (three credits)
Prerequisite: CS210
Surveys computer hardware and software architecture. Focuses on terminology, function, and interrelationship of components as building blocks of information systems. This reference knowledge of computer processors, primary and secondary storage, telecommunications fundamentals, file system and operating system services and computer language systems prepares the student to understand the tradeoffs in systems and network design due to the underlying computer platform.

CS221 Operating System Technology I (one and a half credits)
Prerequisite: CS210

This course explores the basic features and facilities provided by the designated operating system software. The student learns the comparative functionality of this operating system relative to the standard Bentley Computer Configuration system. The student learns all the user interfaces and user controllable options of the operating system and the installation and tuning of drivers, networking options and storage management. Intended for lower level students with minimal computer experience.

CS230 Algorithm and Data Abstractions (three credits)
Prerequisite: CS210
Note: Not open to students who have completed CS245.

This course teaches algorithm development and analysis and the underlying abstractions upon which software systems are based. It focuses on procedural abstractions for controlling the behavior of computer systems and data abstractions for the representation and manipulation of information. The course utilizes the Pascal language as a "live pseudo-code" for the description and exploration of software abstractions.

CS231 Visual Basic Programming Environment (one and a half credits)
Prerequisite: CS101

This course teaches the Visual Basic development environment with emphasis on Visual Basic controls and Visual Basic program structure.

CS232 Visual Basic Application Development (one and a half credits)
Prerequisite: CS231; Prerequisite or Corequisite: CS230

Visual Basic will be used to program a variety of Windows applications. Focus will be on the design and development of complete, polished projects.

CS233 Application Programming I (one and a half credits)
Prerequisite: CS210

This course teaches the environment and function of a particular software development language (e.g. Pascal, C, COBOL, etc.) The course introduces the syntax and compiling environment. Student learn to read source code in the designated dialect and perform program modification tasks in the development environment. Software quality and testing are covered using the testing and debugging facilities of the particular language system.

CS313 Intermediate General Education IT Component (one and a half credits)
Prerequisite: CS210
This course teaches a contemporary IT technology by using a computer based software package. Students are expected to perform operational exercise to gain experience and facility with the particular technology designated for this course section. Students have a broad choice of technology appropriate for students with some technical experience beyond CS101.

CS322 Operating System Technology II (one and a half credits)
Prerequisite: CS221

This course explores the internals and architecture of a complex designated operating system software. The student learns the design history, genealogy and comparative functionality of this operating system relative to the standard Bentley Computer Configuration system. The student learns system interfaces of the operating system and the installation and tuning of drivers, networking options and storage management. Intended for upper level students with extensive computer experience.

CS334 Application Programming II (one and a half credits)
Prerequisite: CS233 of appropriate language

This course teaches the development of production applications using the designated language. It focuses on implementing appropriate control structures, data structures and file structures associated with professional quality applications. The student will be capable of performing all job functions directly related to the programming language technology and understand the strengths and weakness of this dialect relative to Pascal and Visual Basic.

CS340 Computer Networks I (three credits)
Prerequisite: CS220

This course explores the role of communications as an enabling technology for both information systems and the organizations which they support. The course explores issues in the areas of inter connectivity requirements and computer network design in the support of business practices. (Covers communications terminology, principles and standards, and surveys the state of the art in data communications technology and architecture.)

CS360 Analysis, Modeling and Design (three credits)
Prerequisite: CS230 or CS245
Note: Not open to students who have completed CS350

This course begins with business functional analysis and ends with information systems design. Students are introduced to tools and techniques enabling effective analysis, design and documentation of an information system. The student learns formal methodologies that form the basis of modern information systems development practices. Models that focus on the articulation of business functions, integrating process, data and behavioral abstractions form the core of formal methods in systems development and computer assisted systems engineering.

CS361 Data Management with SQL (one and a half credits)
Pre- or Corequisite: CS360 or CS312
Note: Not open to students who have completed CS350

Oracle, a commercially oriented, multi-user database management system, will be used to teach data management. Oracle's SQL*Plus will be used for information retrieval, data protection, and report generation. The use of Oracle for data storage and information retrieval will be presented. This first component will focus on the use and maintenance of data stored in a database management system.

CS362 Computer Aided Systems Engineering (one and a half credits)
Pre- or Corequisite: CS360 or CS312

Visible Analyst Workbench will be employed to teach Computer Assisted Systems Engineering (CASE). VAW will be used to develop and analyze both data and process models. The use of CASE to support Rapid Application Development (RAD) methodologies will be presented.

CS363 Advanced Data Management Technology (one and a half credits)
Prerequisite: CS361

This course teaches the details of database design using the designated database management system. Concurrency control, referential and data integrity, constraints, security, and performance will be learned via development on the database system in the context of data base design and administration. Data backup and recovery, data migration and data conversion features are presented.

CS364 Rapid Application Development Technology (one and a half credits)
Prerequisite or Corequisite: CS360

This course teaches the development of production applications using the designated rapid application development system. It focuses on implementing appropriate control structures, data structures and file structures associated with professional quality applications. The student will be capable of performing all job functions directly related to accelerated life cycle of rapid application development including analysis, design and implementation using the authoring features of the designated environment.

CS401 Directed Study in Computer Systems (three credits)
Prerequisite: Department Chairperson's permission

Permits superior students to study special topics. Allows repetition of credit.

CS402 Advanced Computing Topics Seminar (three credits)
Prerequisite: CIS senior level standing or instructor's permission

Discusses current topics in computing based on readings in the professional literature, guest speakers, and field and individual research projects.

CS414 Advanced General Education IT Component (one and a half credits)
Prerequisite: CS210

This course teaches a contemporary IT technology by using a computer based software package. Students are expected to perform operational exercise to gain experience and facility with the particular technology designated for this course section. Students have a broad choice of technology appropriate for students with extensive technical experience beyond CS101.

CS421 Internship in Computer Systems (three credits)
Prerequisite: (CS312 and CS350) or CS460 and status as CIS sixth-semester full-time major with at least two full semesters completed at Bentley College

Provides an opportunity to develop an extensive project relating computer systems concepts to a specific organization in combination with a work assignment. Involves both full-time employment with an organization and close work with a faculty member. Open to advanced students recommended by a committee of the CIS faculty and the Office of Career Services.

CS440 Computer Networks II (three credits)
Prerequisite: CS340

Building upon the foundations of CS340, this course addresses the emerging technological, organizational and policy issues in data communications networks. Students will learn the techniques of network analysis and design. Networking issues will be discussed with an emphasis on the advanced protocols and standards. Telecommunications policy for the global information infrastructure will be examined from the domestic and international perspectives.

CS450 Object Oriented Technology (three credits)
Prerequisite: CS360
Note: Not open to student who completed Object Oriented Technology formerly CS299.

This course teaches the object oriented paradigm which encompasses object oriented analysis, design, programming and database. By focusing on object oriented modeling the student is prepared to assimilate the widest variety of methodologies and tools applicable to object oriented information system building.

CS451 Object Oriented Programming Environment (one and a half credits)
Prerequisite or Corequisite: CS450

This course teaches the Smalltalk development environment with emphasis on object oriented program structure.

CS452 Object Oriented Application Development (one and a half credits)
Prerequisite: CS451

Smalltalk will be used to program a variety of Windows applications. Focus will be on the design and development of complete, polished projects.

CS460 Applied Software Project Management (three credits)
Prerequisite: CS360 or CS350
Credit is not given for CS395 and CS460

Students learn and experience the process of information systems development through managing team dynamics and performing software engineering project management. Specific topics discussed include the value of different software development life cycles, project management tools and techniques, software process management practices and software quality management practices. This course fuses the student's prior IT and business education preparing them to launch their professional IT careers.

The University of Redlands Honor's Distinction for Bachelor of Science in Information Systems

Dr. Ruth Guthrie, Guthrie@uor.edu
Dr. Donna Schaeffer, Schaeffe@uor.edu
Partick Olson, Olson@uor.edu
Dr. Jim Pick, Pick@uor.edu

University of Redlands, ANWC
1200 E. Colton Ave
Redlands, CA 92373-0999
909 335-4068 fax:909 335-5125

Abstract

This paper describes the University of Redlands honors program for the Bachelor of Science in Information Systems. The honors program was introduced in 1994, to give exceptional information systems students acknowledgment for their intellectual and technical ability. The honors option also gives students an opportunity to work on interesting, company-based problems and a chance to distinguish themselves within their own organizations. The honors program success will continue as we continually improve project quality, currency and relevance. This paper describes the program, gives examples of honors projects and describes ways that the University of Redlands will improve the program.

Introduction

Two colleges and a center comprise the University of Redlands. A residential College of Arts and Sciences houses full-time traditional college students, the Johnston Center has traditional college students that create their own programs and Alfred North Whitehead College (ANWC) offers accelerated programs towards degree completion. ANWC was designed to accept students who have made great progress towards their junior college year. Students are required to have five years of work experience and a minimum of 40 transferable units from other programs. Consequently, information systems students can be very advanced technically and have a great deal of business savvy before entering the Bachelors of Science in Information systems (BSIS) program.

In 1996, ANWC had 200 students enrolled in the BSIS degree program. Students are organized into cluster groups that proceed, lock-step, through the program. ANWC has 3 full-time IS faculty, and 60 adjunct faculty working at 5 regional centers throughout Southern California. The BSIS degree was introduced in 1986, the revised BSIS curriculum was introduced in 1991 (see Figure 1) and the honors option was introduced in 1995. This paper describes the honors program at the University of Redlands and its pluses and minuses regarding the BSIS degree. Specific honors projects, successes and non-successes are given as examples and our plans for improving the future of the honors program are delineated.

At the University of Redlands, every student is required to complete a senior project prior to graduation. Receiving one's degree with honors distinction is not possible without student completion of an honors project. In addition, there are university-wide GPA requirements for honors, starting with a minimum GPA of 3.45. At ANWC, the BSIS degree was offered initially without an honors option. Sadly, extremely bright, knowledgeable students with high grade point averages were unable to graduate with honors because the honors option did not exist. Today, bright students can choose a honors option that recognizes their talents and individuality.

The importance of having a project practicum in our curricula is that it integrates IS courses taken during the program towards a practical, hands-on, business application. Participating in such an endeavor prepares our students for successful information systems development opportunities they may encounter on their present jobs or in the future. The importance of having an honors option is that it recognizes exceptional students and

Figure 1 - Bachelor of Science in Information SystemsCourse Sequence

Course Number	Title	Units	Weeks
	Semester I (Beginning of Program; 28 weeks, 14 units)		
MGTW 310	Philosophical Foundations of Management	6	12
MGTW 325	Management Theory and Practice	3	6
ISYS 314	COBOL Programming Techniques	2	4
ISYS 324	Telecommunications	3	6
	Semester II (30 weeks; 15 units)		
MGMT 345	Management of Organizational Behavior	3	6
ISYS 303	Database Concepts	4	8
BUAD 337	Political and Business Economics	3	6
ISYS 404	Systems Analysis and Design	5	10
	Semester III (20 weeks; 13 units)		
ISYS 488	Applied Software Development Project I	3	0
BUAD 368	Accounting for Managers	3	6
BUAD 332	Business Statistics	4	8
BUAD 461	Financial Management	3	6
	Semester IV (18 weeks, 12 units)		
ISYS 489	Applied Software Development Project II	3	0
ISYS 415	Computer Ethics	3	6
BUAD 470	Production and Operations Management	3	6
ISYS 425	Management and Decision Support Systems	3	6
	Total	54	97

Figure 2 - Generic Project Practicum Final Report Contents

1.0 *Title Page*
- 1.1 Document Identification
- 1.2 Project Title
- 1.3 User Organization
- 1.4 Student Name
- 1.5 BSIS Class and Group
- 1.6 Due Date

2.0 *Table of Contents*
- 2.1 Table of Contents
- 2.2 List of Figures
- 2.3 List of Tables

3.0 *Introduction*

4.0 *Project Description*
- 4.1 Project Contract
- 4.2 Project Management Schedule
- 4.3 Feasibility Assessment
- 4.4 Detailed Study Report
- 4.5 Project Requirements Document
- 4.6 Feasibility Analysis Report
- 4.7 System Proposal

H.0 *Literature Survey (honors only)*

5.0 *System Requirements*
- 5.1 Essential Data Model
- 5.2 Essential Process Model
- 5.3 Network Model (optional)
- 5.4 Organization Chart (optional)

6.0 *System Design*
- 6.1 Implementation Data Model
- 6.2 Implementation Process Model
- 6.3 Project Repository

7.0 *Detailed Design*
- 7.1 State Transition Diagram
- 7.2 Input/Output Screens
- 7.3 Prompt and Help Dialogue
- 7.4 Structured English
- 7.5 Sample Reports

8.0 *Implementation*
- 8.1 Code Listing
- 8.2 User's Manual

9.0 *Operational Results Reported*
- 9.1 Signed User Acceptance Letter

10.0 *Conclusions and Recommendation*

gives them a forum in which they are challenged and creative. They can show their technical savvy and can potentially gain professionally from their work at the University of Redlands.

Overview of the Honors Program

Timeline and Placement in the Program - Students begin thinking about their honors project during their course work in Systems Analysis and Design which is half-way into the BSIS curriculum. The instructor for this and for the two project practicum courses which follow is usually the same person. This instructor is effectively the chair of their project. Students usually select a project from their work though this is not a requirement. Students may also select projects from public or private agencies, governmental agencies or non-profit organizations, etc. The project must apply to a business information management problem.

The BSIS program at ANWC is an accelerated program of 97 weeks (4 semesters). The BSIS curriculum shown in Figure 1 is equivalent to a junior and senior year. Students take their first two years of foundation and general education courses elsewhere and transfer those units into the BSIS program. There are two course pre-requisites for entering: an introduction to IS course and a course in a third generation language.

The project is completed over the last two semesters, meeting eleven times throughout the 38 week period to discuss systems development, progress and turn in project deliverables at the appropriate milestones. Students refer to the BSIS Project Guide, a handbook that is based on the systems development life cycle (SDLC). Often, more meetings with faculty and cluster groups convene according to individual student initiative. During the project practicum courses, students develop their projects using structured systems development methodology. While students follow a waterfall model, they are strongly encouraged to prototype their systems and gain user feedback A typical project practicum table of contents is shown in Figure 2.

The Honors Option - By the seventh week of Systems Analysis and Design, a student must declare their intent to do an honors project (Figure 2). This amounts to the student sending a memo to their instructor and the director of the IS program with a 4 page proposal of their project. To graduate Cum Laude, Magna cum Laude or Summa Cum Laude, the student must have a GPA ranging from 3.45-3.64, 3.65-3.84 and 3.85-4.00 respectively. Students take more courses prior to completion of their honors project. If they fail to keep a high enough GPA, a degree will be awarded without honors.

There are three areas in which the honors option requirements exceed those of the traditional project. Honors students are required to conduct a formal literature survey. The survey requires students to look beyond the context of their organization in development of their system. They examine current academic research, industry trends and technological developments that relate in their project. The literature reviewed must be scholarly and integrated into 3 or 4 sub-themes. The literature survey becomes a formal chapter in their proposal. Often this stimulates students to look at the 'bigger picture' regarding information systems among industries and to gain research experience in developing systems.

Placing their project into a broad, industry context often compliments the second, subjective requirement: the honors project must have heightened relevance and complexity in comparison to a non-honors project. This requirement is easiest for student to meet. A person who chooses the honors option has exceptional grades, is highly motivated and is interested in developing a system that has relevance and impact. It is an issue of personal pride, not degree completion.

The third, and most difficult, requirement is that a creative/innovative element be present in the honors project. Students are not accustomed to thinking of creativity in the context of information systems. However, it can be present in all aspects of systems development. Perhaps the solution to the business problem is the creative element in the project because it represents a new way of thinking for the organization. Often highly creative solutions do not require complex technical systems.

It is challenging to the honors student to propose the creativity component, since creativity can only be pre-planned to a limited extent (Couger, 1994). The project faculty chair must guide the student to define the broad area when creativity is sought, rather than demanding creativity at the start.

A committee of three faculty members (usually the project practicum instructor, a full-time IS faculty member and an adjunct faculty member) formed by the BSIS program director (Dr. Donna M. Schaeffer during 95/97) meets after the project class is completed to participate in an oral defense of the honors project. If they pass on the defense and the student has maintained a high GPA, an honors distinction will be shown on their degree. Recognizing that the honors project requires greater guidance, a modest stipend is provided to the chair committee members of the honors project.

Honors Experiences

The following items are short examples of BSIS honors projects undertaken at ANWC. The projects are interesting in that they show the depth an breadth of our students ability and they also indicate that, as in business, not all information systems are successful. It is also of note that many honor projects are proprietary - a flattering indication of their success.

- Geographic Modeling of U.R. Undergraduate Applicants (1994) - This project attempted to perform geospatial analysis of CAS applicants relative to census regions. The honors option for the project was never completed due to inadequate software and imperfect access to data. The site contact on campus was undergoing downsizing which limited the accessibility of key users. The lessons learned were in determining project scope and software selection and, in the importance of data accessibility to any project. The project was attempted in ArcInfo for Sun stations and PC-based Map Info.

- Electronic Library (1995) - This project streamlines an automated paper processes related to customer service at a phone company. Annual costs savings to the company is expected to exceed thousands of dollars not including productivity, accuracy and customer satisfaction gains. The creative element of this project is that it linked two organizations within the company and used brainstorming sessions to guide requirements. the student received a company process improvement award from their company. The project was developed for Windows in MS Access.

- Critical Dates in Litigation (1995) - This project merged information from federal, state and county rules databases with lawyer and client databases to create a decision support system that would manage critical dates for litigation. Previously, dates were difficult to calculate and often missed. While many law firms are "wired" regarding information searches, few use technology for case management. The project also required incredible flexibility to change with legislative rulings and to grow with the firm. Currently, the project is in use at the student's firm and will be a module in a greater program to manage more aspects of the firm's business. The project was developed for Windows in Turbo Pascal.

- Intelligent Project Support (1996) - This project created a project analysis, data repository for competitive querying. This made all company project information and some competitor information available on-line to sales, marketing and executive offices. The project is now fully implemented at the company and is highly proprietary. The project was implemented in Microsoft Access.

- Integrated Order Entry System (1995) - This project attempted to re-engineer an existing process and create a network version of the order entry system across a very large corporation. The project failed because it was too ambitious for the short time period allowed. This indicates the importance of defining the scope of one's work and the important role of the project advisor in helping a student find a task that is possible.

- On-line Help using MS Hulk (1996) - This project implemented on-line help for a local company in a highly innovative way using the Microsoft Hulk compiler. The project had problems during the requirements phase but became a great success. The student developed application was purchased by the company.

Selecting an Honors Project

Selecting a honors project is much like selecting any project (Figure 3). The size and scope of the project must be small enough to be completed in 38 weeks, the student must have access to the technology and the student is more likely to succeed if they are interested in the topic. If the student can select a project from work, they already have access to supporting resources and technology and have the support of their company in the successful completion of the project. Students at Redlands often find technical assistance from expert members of their learning clusters. This is a powerful way that clusters can work to facilitate teamwork in a more natural work setting.

The project instructor is also key to project success. A knowledgeable project instructor will often foresee pitfalls and problems, guiding students towards successful completion of the project. As in industry, project change as they evolve. A project instructor must be flexible as projects change yet, be rigid in making sure the project comes to completion.

An honors project differs from a typical student project in that it is broader in its implications and has some creative, innovative aspects associated with it. Systems

Figure 3 - Choosing an Honors Project

	Logistics	Technology	Human
What makes a good project?	• Size and scope of the project	• Access to technology	• Student interest • Work support • Cluster support • Instructor support
What makes a good honors project?	• Systems thinking • Unleash creativity	• Use of new technology • Use of tested/supported technology	• Emphasize user communication • Exceptional students • Time

thinking begins during the literature survey where students look across industries and fields of research. A database project may gain insight from fields like sociology, law and human computer interaction.

Sometimes students miss that an innovative project is novel and creative. Often creativity occurs during the projects evolution in unexpected ways that cannot be foreseen early on. Students can define areas of potential creativity by focusing on specific areas of the life cycle or by emphasizing productivity, fostering organizational communication or streamlining information processing. At the onset, students are often reluctant to exercise their creative minds. The challenge of the honors project often brings forth talents they were unaware they had.

Projects that use new technologies are extremely good for honors projects because the applications and ideas associated with them are emerging. However, a student needs to temper innovation with how well these technologies are supported. Frequently, honors students call vendors to inquire about product specifications and problems.

The user is critical to honors project success. At Redlands, user/customers are required to sign project acceptance letters. Projects have less problems during the life cycle when good communication with users is established early in development. This is a lesson we cannot emphasize enough to our students working as IS professionals.

Lastly, honors is not for everyone. The time requirement is very stringent. It takes an exceptional effort from an exceptional student to successfully complete an honors project. We hope, that the broad project goals and systems thinking help prepare an honors student for greater success in their careers and motivate them to continue their education towards a graduate degree.

The Future

Changes in the project honors option will include addition of the evolutionary spiral model (Boehm, 1988) and building on current unique opportunities that technology and our faculty offer. Using the waterfall model is attractive because it is easier for instructors to manage and it teaches concepts that are applicable in any software development methodology. It is unattractive in that its application is best for well defined, highly structured, stable systems. These systems typically exist already and there is little innovative work for a student to do. Furthermore, spiral software development is rapidly becomming the industry standard.

At Redlands, we have several faculty (full-time and adjunct) actively involved in GIS programs. GIS honors opportunities are enhanced by a campus partnership with ESRI, the world's leading GIS software vendor, located in Redlands. Honors projects in GIS promises to be an exciting area for us to encourage.

The growth of the world wide web with applications in decision support, marketing and on-line help may provide some interesting honors projects as well. As new Web technologies like JAVA develop, we have an opportunity to keep ourselves and our students well versed in leading edge technology and attractive for industries emerging information systems fields.

Summary

The honors option in the BSIS program at University of Redlands started in mid 1994. It is clear that honors is not for everyone. Honors requires extra student work. Adult professional students must balance that extra load against the competing demands of full-time jobs, family and community commitments. Some honors students find they don't have the motivation to see it through. Attrition is the inevitable outcome for some, as is true in any honors program. This points to the project faculty and program director not encouraging less promising students. Students who drop honors need to be supported in their self esteem including accomplishing the BSIS degree and a normal project.

Among the advantages of honors in an IS program are the following:

- Honors challenges bright, techno-savvy students to achieve at their best.
- Honors stimulates student potential for creativity. the kindled creativity can be a career plus. Couger has proven that a benefit of IS creativity is sharply heightened productivity (1994).
- Students apply the goals/knowledge learned in the BSIS curriculum in an expanded integrative way, exceeding the capstone experience of a normal project.
- Student increase their depth of academic knowledge through reference to the literature.
- The student interaction with faculty is enriched and a broader student-faculty network occurs.

Some future anticipated benefits include potential for student publication. This would help IS graduates strengthen their resumes for career advancement. Honors work awakens scholarly capabilities that can lead to graduate school entry and accomplishments. Also, students will often greatly benefit their companies through innovative system advances. IS programs may in turn benefit from corporate reciprocations (Pick and Schenk, 1995)

On balance, IS/Honors at Redlands has enriched the student learning and tapped the hidden creative potential of the exceptional student.

References

Boehm, B., and Frank Belz, F. (1988) Applying Process Programming to the Spiral Model. In *Proceedings of the 4th International Process Workshop*, May.

Couger, J. Daniel. (1995) Creative Problem Solving and Opportunity Finding. Denver: Boyd and Fraser Publishing Co.

Pick, J. and Schenk, K. (1995) Development of An Information Systems Curriculum for Non-traditional Students, *ISECON '93: Proceedings of the 10th Information Systems Education Conference.*

Linking Accounting and Computer Instruction: A Teaching Innovation

Marilyn L. Wilkins, Ed.D
Department of Business Education and
Administrative Information Systems
Eastern Illinois University
Charleston, Illinois 61920

Lola W. Dudley, Ph.D.
Department of Accountancy and Finance
Eastern Illinois University
Charleston, Illinois 61920
cfmlw@eiu.edu or cflwd@eiu.edu

Abstract

This paper describes an experimental project in which a Principles of Accounting class and an Introductory Computing class were linked. The same students were enrolled in both sections. During the course of the semester, the students completed several common problems for the two courses. The professors coordinated their instruction. The linked course was considered a success by both students and professors.

Introduction

Many of the predictions regarding the widespread availability of computers among college students have become today's reality. Along with the availability of hardware and software has come the responsibility for business educators to assure that graduates are able to use these electronic tools. If graduates are to develop the computer requisite, computer applications must be integrated into many business-core courses.

Business educators have long recognized that a logical place to begin to develop students' ability to use computers is the introductory accounting course. The benefits expected to result (Togo and McNamee, 1995) include:

- Reduction of the computational burden
- Improvement of analytic skills
- Promotion of analytic understanding of accounting
- Aiding learning process through attention direction
- Facilitates learning about the integrative nature of accounting
- Targets report formats and their data requirements
- Allows for a decision-maker approach, instead of a preparer approach

For years, though, accounting faculty have complained that introductory-accounting students do not have enough skills with computers, particularly spreadsheets, to use them in solving accounting problems. In 1995, Togo and McNamee reported that only 42 percent of faculty use microcomputers in introductory financial accounting courses. The need to teach computer skills presents a great impediment to the use of computer applications in this course because it decreases the time available for teaching and learning **about accounting**.

At the same time, faculty teaching the introductory course in computers have been complaining that students have little business or accounting knowledge to use in solving problems in microcomputer applications. As a result, the computer applications covered in this course tend to be "canned," which has a negative effect on student attitude and does not permit students to see the relevance of computer-based techniques to many business problems.

The Linked Courses:

Principles of Accounting & Computer Concepts and Business Applications

To address both of these problems, we linked ACC 2100 (Principles of Accountancy 1) and AIS 2100 (Computer Concepts and Business Applications). The same students were enrolled in the linked sections and the courses were coordinated so that students learned the needed spreadsheet skills as required to solve accounting problems. With students taking both courses, the accounting professor was able to place more emphasis on the problem-solving and decision-making aspects of accounting and less on the mechanics of accounting. By using spreadsheets to record transactions. students were able to immediately see the effects of changes in accounts on the financial statements. The computer instructor was able to teach students how to create their own templates and models to solve accounting problems. This, in turn provided students with a much greater understanding of accounting than textbook models. which often require students only to enter numbers, would be able to provide. Students could **see** a practical use of the spreadsheet applications they were learning.

These linked courses also address one of the main criticisms leveled at business schools in recent years: The lack of interdisciplinary teaching and learning. By linking these courses, a great deal of cross-disciplinary integration took place. In most cases, the new spreadsheet concepts were covered immediately before these skills were needed in the accounting course. In some cases, though, the computer skills were **used** by the accounting professor and the students were then taught them in the computer course.

An incident which demonstrated the motivational nature of this project occurred immediately after the accounting professor used a spreadsheet in the accounting class to illustrate the effects of adjustments on the balance sheet and income statement. The students came to the computer class and asked the computer professor to begin teaching the spreadsheet software so they could solve the accounting problems like their accounting professor had.

Throughout the semester, students completed several common assignments which "linked" the two classes. These common assignments were submitted to both professors for evaluation in their respective classes. The accounting professor evaluated the assignments for accounting concepts and the computer professor evaluated the assignments for spreadsheet concepts.

Common Assignments

The students used a worksheet to enter changes in accounts while keeping the accounting equation in balance. Students solved a problem from their accounting text using spreadsheet software. This problem required the use of cell references and the built in addition function. The need to keep the accounting equation in balance made it much easier for students to see the operation of the double-entry system of recording transactions.

The students first used a spreadsheet template to adjust accounts for end-of-period accruals and deferrals. The spreadsheet then computed an adjusted account balances. From the adjusted trial balance, students prepared the Balance Sheet and the Income Statement in another section of the spreadsheet by making cell references to the figures in the adjusted trial balance. This problem illustrated the use of the spreadsheet as a template and the power of recalculation and cell references to avoid re-typing figures. The students reused this template three times in the accounting class, further reinforcing the efficiency of using templates.

The students performed trend analysis of financial data from annual reports. Each student in the accounting class worked with a different firm's annual report. Students calculated trends of specified figures on their firm's financial reports. This problem illustrated the use of absolute cell references and using formulas in the spreadsheet. In the accounting class, students compared "their" firms' trend figures with those of their teammates and selected one of the firms as an investment.

Students then prepared graphs illustrating the trends they had found in the previous assignment. Obviously this was a practical application of creating graphs and presenting information. In the accounting class, this assignment provided the background for a discussion of the charts and graphs found in annual reports and the ways that charts can be used both to uncover and to hide information.

Students prepared common-sized financial statements for "their" firm. This assignment illustrated copying formulas and using absolute cell references for efficiency. In the accounting class, students used the common-sized statements to analyze key relationships between financial statement items.

In the final common assignment, the students used IF statements, nested IF statements, lookup tables to calculate payroll deductions and take-home pay. Adjusting entries were made in the accounting class using these numbers.

Reactions of Students

At the end of the semester, students in two computer courses taught by the computer professor were asked to assess the courses. Students in the linked computer course and students in a regular computer course completed questionnaires about their opinions of usefulness of the computer for accounting. Additionally, the students in the linked course were asked to assess the linking of the courses. There were 22 students in each section. Of all the students responding, 72 percent were currently enrolled in accounting, 19 percent had taken college accounting previously, and 9 percent had had no previous accounting. Fifty-seven percent had taken a computer course in high school, 23 percent had taken a college computer course previously, and 21 percent had not.

The results of comparisons of students' opinions of the usefulness of the computer in accounting are shown in Table 1. It appears that students in both classes believe that the computer will help them in future courses and would save time. However, students in the linked class have a significantly different opinion about the computer being more accurate than students in the regular class. It appears that perhaps the students in the linked class have a more realistic appreciation for the accuracy required of the user when using spreadsheets for accounting.

Only students enrolled in the linked section were asked to evaluate the effectiveness of linking the courses. Their opinions are shown in Table 2. Students were most positive about their confidence in using the computer in accounting and that the instruction they received prepared them to use the computer in accounting.

Table 1
Comparison Between Regular and Linked Classes
(Mean Response)

1=Strongly Agree 2=Agree 3=Neutral 4=Disagree 5=Strongly Disagree

Question	Regular Class	Linked Class	Both Classes
I believe using the computer will help me in the future in accounting classes.	1.77	1.68	1.73
I believe using the computer will help me in the future in other business courses.	1.31	1.45	1.38
Using the computer would save time in solving accounting problems.	1.72	1.63	1.68
Using the computer would be more accurate when solving accounting problems.	1.50	2.05	1.78 $p<.05$

Table 2
Opinions of Students in Linked Courses

1=Strongly Agree 2=Agree 3=Neutral 4=Disagree 5=Strongly Disagree

Question	Means
The computer was useful in learning accounting.	2.68
My confidence in using the computer in accounting improved during the semester.	2.18
Using the accounting worksheet approach helped me understand the impact of transactions and adjustments on the balance sheet and the income statement.	2.45
The instruction I received in computers adequately prepared me for the computer assignments in the accounting class.	2.22

Reactions of Faculty

The computer professor felt that the innovation added a realistic dimension to the teaching of the computer applications course. Students were able to **apply** skills to practical problems simultaneously with **learning** the skills. The students were able to see the relevance and value of the computer skills being taught. The use of templates for accounting problems was an excellent method for teaching the usefulness of spreadsheet templates because students were able to actually "reuse" a problem for credit. Students appeared to be more motivated to use spreadsheet software than students in regular computer class.

The linking also made the computer professor aware of the "chasm" that exists between learning computer skills and applying them in business courses. The problems used in computer courses tend to be highly structured and getting the "right" answers usually depends on using the correct computer (spreadsheet) procedures. Applications, on the other hand, require not only knowledge of the procedures, but also the ability to create spreadsheets using unstructured data and then to analyze the information.

The accounting professor stated that clear benefits resulted from linking the accounting principles course with the introduction to computers course. Although a computerized ledger package was not used, use of the spreadsheet throughout the course considerably reduced the computation burden for the students. They were able to see the effects of transactions on the financial statements without having to go through all the procecures (journalizing, posting, taking a trial balance, recording adjusting entries, preparing an adjusted trial balance, and preparing the financial statements) necessary in a pencil-and-paper approach.

For the first time, the accounting professor was able to construct the course from a decision-maker approach instead of a preparer orientation. This approach makes the course much more relevant to non-accounting majors, who make up the majority of principles of accounting students, while providing valuable insights to accounting majors.

The use of spreadsheets made it feasible for students to analyze information from actual annual reports. Without the spreadsheet, this analysis would have been much more limited. Such an analysis is very valuable in helping students understand the uses and limitations of financial accounting information.

Finally, linking of principles of accounting with introduction to computers made it possible for the accounting professor to use the computer to help students **learn accounting**. All the software packages used previously had focused primarily on the computer applications and provided the students almost no additional understanding of accounting.

Conclusion

The linked course was considered a success by both the students and the professors. Both professors are enthusiastic about the efforts. The most difficult challenge was to schedule the same students into two different classes. The linking is being repeated this semester so that additional data can be gathered.

References

Accounting Education Change Commission. (1990, Fall). AECC Urges Priority For Teaching In Higher Education. *Issues In Accounting Education*, 5(2), 330-331.

Cherrington, J.W., Denna, E., and Romney, M. (1995, May). An Innovative Accounting Information Systems (AIS) Course. *1995 IRMA Conference Proceedings*, 3-11.

Franz, D., & Huang, J. (1992, July/August). Alternative Spreadsheet Applications: Student Learning And Attitudes. *Journal Of Education For Business*, 67(6), 377-382.

Hiltebeitel, K., Jones, S., & Harmon, W. (1990). Student Attitude Shifts Attributable To Accounting Microcomputer Projects. *Kent/Bentley Journal*, 56-64.

Loyland, M., & Hawthorne, J. (1993, Spring). A New Teaching Approach For The 1990's? The Linked Course: Principles II And Composition II. *Accounting Educators' Journal*, 5(1), 122-132.

Togo, D., & McNamee, A. (1995, Spring). Computer Integration Into The Accounting Curriculum: Learning Benefits, Problems, And Guidelines. *Journal Of Accounting Education*, 13(2), 149-158.

IS Professionals Suggest Courses For Undergraduate Majors In Computer Information Systems

George W. Morgan
Computer Information Systems
Southwest Texas State University
San Marcos, Texas 78666
(512)245-3229
email: gm07@business.swt.edu

and

Mayur R. Mehta
Computer Information Systems
Southwest Texas State University
San Marcos, Texas 78666
(512)245-3223
email: mm07@business.swt.edu

Abstract

This paper presents the results obtained from a meeting and workshop held with industry advisory council members to assess the currency of an undergraduate Computer Information System (CIS) program. The collaborative efforts between academia and industry for identifying and recommending the CIS curriculum courses are reviewed. The resultant set of core courses are summarized with a follow up on recommended elective course content. Revised IS curriculum generated from a strong bond between academia and industry has the potential of moving IS education into the twenty-first century.

Introduction

The end user's status has given expression to many of the dynamic changes associated with the computer technology evolution over the decades. In the sixties and seventies, the user had to rely on experts and then specialists. They were limited to either a centralized computer room or relegated to use one of a series of terminal rooms for subservient access and/or dependent support. Then, in the eighties, the user gained some independence using desktop computers while running generic applications; they were becoming functional end users. The nineties has given us mobile workgroups using component applications with communication links to potentially every other corporate user and beyond. End user empowerment and the freedom to effectively manage the business has brought about a paradigm shift (Tesler, 1995) in the way industry and academia view the role of information systems.

Industry IS managers have found it difficult to recruit people who possess the newer distributed computing skills (Jenkins, 1994) needed to mobilize the end user groups. Academia must first define and implement the essential instructional models before a company can expect to acquire skilled employees able to keep pace with the shift toward client/server technologies (Goff, 1995). Many concerned professionals are coming to the conclusion that IS managers need to become more involved in the educational system (Currid, 1992). That involvement can be vital to academia; not only is industry a major source of information about technology based change, it remains the primary employer of IS graduates (Pick, 1995). Revised IS curriculum generated from a strong bond between academia and industry has the potential of moving IS education into the twenty-first century.

A growing dissatisfaction with the existing IS curriculum has been developing within resent years (McGuire, 1995). Industries restructuring and reengineering efforts have shifted the balance of computer expertise in favor of new technological skills that were non-existent just a few years ago. Academia is playing catch up with new technology and industry is beginning to challenge its long-standing corporate education and recruitment practices. Now faced with the emergence of high-speed networking, client/server has become the dominant architecture for new systems development. A concerned assessment of the IS program by academia has barley begun to impact on the student's need for new technological skills (Welch, 1992; Gillin, 1995). A framework for meaningful curriculum relationships can bridge the gap between academia and industry quickly and effectively.

Curriculum In Progress

Networking, client/server architecture, business object analysis and design, relational databases, and advanced tools for application development are clearly the future for computer information systems. Programmer-analyst and network technical-analyst have become the dominant job titles available to the prospective graduate and should be considered when assessing further changes within the curriculum. While the temptations exist for separating network related skills from those of programming, an integrated and spiraled approach for teaching client/server concepts is recommended (Robbert, 1995). To determine what combination of courses would work best for our students, we planned and conducted a workshop to be attended by professionals from the IS industry. This assessment process was instrumental in providing needed directions for restructuring the curriculum.

Industry advisory council members for the affected CIS department were placed into separate workgroups for the assessment process. Each workgroup was asked to pool topical ideas about course content and come up with a CIS curriculum that would provide a graduate with the computing and IS skills necessary for the year 2000. No restrictions were placed on the number of courses that a proposed program might contain. The individual groups worked on producing a list of (1) required courses, (2) suggested elective courses, and (3) other courses of value. The blank sheet approach was used for the curriculum development portion of the meeting. That is, council members were not allowed access to any university or departmental reference material. they were to assume that no such material existed.

On the basis of the advisory council findings, several conclusions concerning the CIS degree plan were drawn. First, foundation knowledge of systems analysis and design, database management systems, team and project management skills should be mastered by the student. Second, the new graduate should possess the technical skills needed to develop and implement simple to complex multi-user microcomputer networks. Hands on experience is expected in several areas. These include competency in: (1) computer hardware and software components, (2) network management and interconnectivity, (3) end user computing, (4) programming, (5) three-tier client/server software development, and (6) the full integration of multi-user systems across horizontal levels of business.

CIS Core of Courses

A preliminary core degree plan for the CIS program included four major component areas containing seven required courses (Figure 1). This arrangement of courses would use twenty-one (21) credit hours for the CIS core, while limiting the elective courses to a maximum total of nine (9) credit hours. The core establishes a sequence of courses that are compatible with the acquisition of programming and network related skills.

Client-based programming and server-based programming courses offer training in the construction of the physical documents and business rules needed to connect the end user to the front end of the client/server network. Graphical builders and procedural

Figure 1: CIS Core of Courses

Capstone: Projects in Distributed Computing		
Server-Based Programming	Data Base Management Systems	Client/server Architecture
Client-Based Programming	Systems Analysis and Design	Local Area Networks

programming languages are learned. Desktop GUI applications development provides an event driven environment for the end user while the business rules components can run on any machine and are usually written in a language suited for extensive computations and manipulation of records. The two courses offer an introduction to procedural and event-driven programming while providing intermediate skills in data structures and advanced file processing.

Back end database servers require a different way of thinking about database languages. At some point queries and database request have to be expressed in ways that the database manager can interpret and respond to them. Medium-sized to large applications are written using two or three different languages; each layer in the three-tier client/server applications development model may require its own language. The graphical front end coupled with rules based components interacting with the database back end will be the norm. By far the most dominant language associated with the database function is relational SQL while the more tricky algorithms are expressed using C++. The basic concepts underlying the definitions for the database, data flow, and processing required by the enterprise to do business can be found in the fundamentals for systems analysis and design. The benefits of cooperating components and emerging object oriented methodologies in the three layer application architecture are featured.

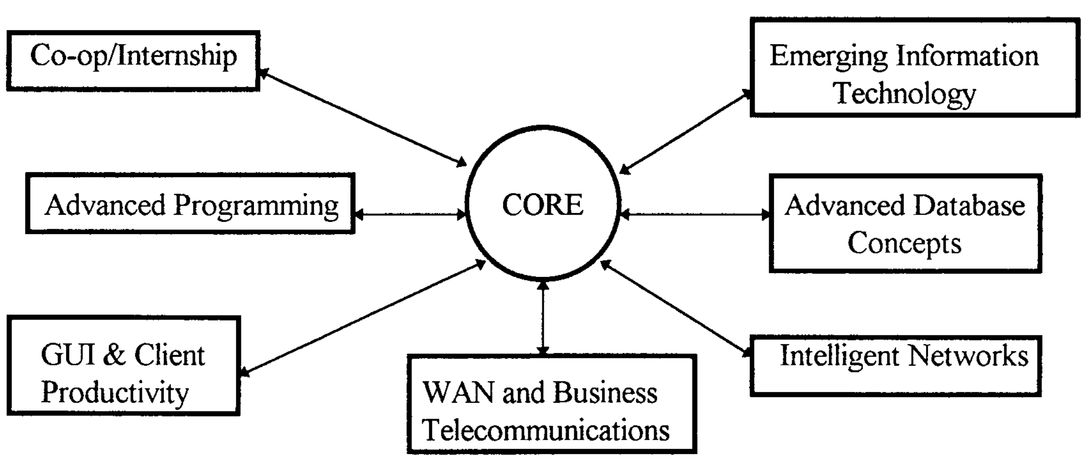
The third major component of the degree plan includes the study of local area networks and client/server architectures. The technical shift to distributed systems affords the student with an opportunity for course work in networking (multiprocessor server) and in emerging client/server methodology, models and the tools used in graphical development environments. The student will gain hands on experience with information technology, servers, network operating systems and management software, middleware, and professional applications development tools. The configuration, installation, and implementation of small to medium-sized local area network systems complete the technical skill's component of the degree plan.

The core degree plan's capstone course is projects in distributed computing. The purpose of the course is to provide the student with an opportunity to interact in a team environment in order to successfully deploy system software across several departments using the three-tier client/server architecture standards. Systems integration, interpersonal issues, and project management are the sub areas of primary emphasis while focusing on the fundamental analysis, design, and implementation of a complex multi-user system. The solution of real client problems is encouraged; however, simulated projects may be utilized when necessary. Outside resource personnel (project steering committees) are used to coordinate, evaluate, and critique each team member at the conclusion of different project phases.

CIS Elective Courses

The advisory council workgroups suggested seven courses as optional areas of study by the CIS major (Figure 2). The elective courses follow the breakdown of core courses into four major component areas - application programming, enterprise modeling, network implementation and industry training. These courses are arranged so as to offer the student additional exposure to

Figure 2: CIS Elective Courses



new technologies, advanced programming techniques, GUI design, advanced database concepts, intelligent network environments, enterprise wide networking, and co-operative/internship arrangements with existing IS intensive enterprises.

Each elective course should be offered as an extension to advanced learning in end user computing, programming, and network operations. A list of recommended elective courses begins with co-op or internship opportunities for juniors and seniors that will offer an exposure to actual business experiences in CIS. An advanced programming course would provide training in an object oriented language that has a focus on object principles and class library design. A strong foundation in object knowledge will let the student move to other object oriented languages (such as smalltalk, C^{++}) and professional client/server development tools based on these OO languages (such as VisualAge, VisualWorks) more easily.

End user computing can be enhanced by offering an advanced client productivity course. Students would gain experience with workstation decision support tools and more effective GUI development using human factors engineering. In addition, they would gain experience in designing applications that integrate a myriad of desktop decision support tools to provide seamless access to enterprise data.

WAN/business telecommunications and intelligent networks are courses that will provide a continued focus on enterprise wide decision making, connectivity, multi-tiered client/server architectures, and automatic information agents. Each of these lead into a relationship with emerging information technology and advanced database design. Multi-media servers, groupware, middleware, video conferencing, and telecommuting are a few of the concepts that are being introduced. Advanced database concepts are expected to offer an exposure to distributed database systems, object oriented techniques, database computers, and intelligent systems. Technical skills can be applied to the logical design of systems that enhance the use of data warehousing, knowledge worker environments, and on-line analysis.

Conclusion

The purpose of this paper was to present an undergraduate Computer Information Systems degree plan that is firmly established in a strong working relationship with industry. The new curriculum format would prepare majors for the increasingly technical job markets of the coming decade. We found that the most important question a school or academic department of CIS can ask is, what curriculum will give our majors a competitive advantage? The answer was instrumental in closing the perceived educational expectation gap with industry (Hensel, 1995). It may also justify increased efforts at working together in ways that can directly impact the acquisition and training of emerging technologies. Without cooperation, shrinking academic budgets may mean a more limited use of appropriate technology in the classroom. With cooperation, the academic program may be able to keep up with rapid changes in technology more effectively, thus, providing the graduate with a current toolkit of marketable CIS skills. In this light, the curriculum is seen as a bonding agent between academia, industry and student.

References

Currid, Cheryl. "IS Curriculum Continues to Evolve at the Corporate Level." InfoWorld, June 1992, 64.

Gillin, Paul. "Retrain, Don't Trash." Computerword, May 1995, 36.

Goff, Leslie. "Retooling Success." Computerword, March 1995, 132.

Jenkins, Avery L. "Overhaul Success." Computerword, July 1994, 98-99.

Hensel, Mark. "Closing The Gap Between Academic Degree Programs and Industry Expectations." Proceedings: ISECON '95, November 1995, 70-76.

McGuire, Eugene G. "An Undergraduate Degree in Multimedia Design & Development: Collaboration Between the Departments of Computer Science and Information Systems and Art." Proceedings: ISECON '95, November 1995, 17-24.

Pick, James B. and Schenk, K. D. "A Framework for Information Systems Programs and Business Relationships." Proceedings: ISECON '95, November 1995, 62-69.

Robbert, Mary Ann. "A Model for Integrating Client/Server into the Undergraduate Curriculum." Proceedings: ISECON '95, November 1995, 269-273.

Tesler, Lawrence G. "Networked Computing in the 1990s." Scientific American, Special Issue 1995, 10-21.

Welch, Douglas E. "Colleges and Companies Must Alter Strategies for the Changed Job Market." InfoWorld, October 1992, 51.

Integrating the Internet and WWW into Information Systems Curricula

Nancy Thomson, Ph.D.
Department of Computer Science/Information Systems
Northwest Missouri State University
800 University Avenue
Maryville, MO 64468-6001
thomson@acad.nwmissouri.edu

and

Amita Goyal, Ph.D.
Information Systems Department
Virginia Commonwealth University
1015 Floyd Avenue
Richmond, VA 23284-4000
amita@saturn.vcu.edu
http://isy1.isy.vcu.edu/isydept/faculty/amita/goyal.html

Ubiquitous access to information has leaped the boundaries of information systems and business processes and become embraced by the clients and constituents of profit and non-profit firms. Institutions of higher education must bear the burden and react successfully to the challenge of educating and preparing students in an holistic manner to become Web-knowledgeable professionals. The time to create and sustain expertise in effective development, design, and utilization of this tool/technique must become an immediate and dynamic piece of information systems education.

Introduction

The growth of the worldwide Internet has spawned many advances in information technology with educational research institutions often taking a leading role in championing the efforts. As the Internet flourishes in its second decade of existence, it has become extremely visible to the business community as well as to the general public. While the substantially increasing use of electronic mail has greatly contributed to this phenomenon, much of the expanse has been a direct result of the invention of the World Wide Web (WWW).

The WWW, designed in 1989 by Tim Berners-Lee is a user-friendly, graphic-based system for accessing Internet resources. The WWW or "Web" consists of a Web client, a common hypertext publishing language, a common network communications protocol, and a Web server.

The WWW has transformed much of the previously arcane methods of distributed information retrieval into a user-friendly, inviting graphical user interface (GUI) supporting mouse clicks and multimedia. This has resulted in unprecedented popularity and usage -- both in the number of people accessing the Web and in the number of companies, organizations, and publications establishing Web sites. The Web's explosive growth rate (Conhaim, Chapin) has surpassed the growth rate of even the Internet itself. Data indicates the use of the WWW has doubled every few months, which represents twice the rate of overall Internet growth (Semich). Corporate America has seized this technological opportunity as a means of competitive advantage and created a multitude of 24 hour international store fronts, complete with Web-based advertisement, literature disbursement, and procurement.

The rapid expansion of the Internet with World Wide Web (WWW) functionality has proliferated to a large extent by the phenomenon of widespread connectivity and file transfer capabilities as well as the freeware or shareware (Sutton) which proliferates there. Access to a myriad of freely and easily downloadable resources (some of outstanding quality), defy the typical market forces of the standard money-based economic systems. The idea of something for nothing always creates interest, even if the "something" is of only marginal value.

This price tag has provided opportunities for schools and universities who already have substantive computing resources to quickly absorb the concept of uninhibited access to information. All levels of education can now, at nominal cost, bestow Internet access upon students and faculty. From kindergarten through graduate students, this newly developed tool/technique has given rise to novel methods for expedient and relatively effortless communication and information access.

This ease has a down side, however. With literally anyone capable of availing material via the Web, the information filters which we find in existing public forums and information providers is simply non-existent. This implies that for organizations who wish to have their sites respected as reliable, responsible sources, good design and control of development, deployment, and interaction is necessary. With the view of current curricula focused on using the Web as a source of information, there is a dearth of sincere conceptual development and structured guidelines for effective creation of this resource as a critical component of information systems (IS).

In this paper, we propose the prudent expansion of information technology (IT) curricula to respond to capricious technology which has provided the catalyst to introduce Web technology design and concepts into undergraduate education. We consider the subliminal integration of the technology into existing information systems curricula as well as the creation of a new course to address the well established need.

Discussion

This period of Web development is not significantly different from other IS arenas that have gained prominence in the past. We have seen the rise and fall of numerous programming languages, techniques, and procedures based upon volatile technology. The differentiating fact about the Web, however, is its unprecedented expeditious recognition and adaptation by the general public and visible businesses such as the commercial media. When we watch the local or national news and see the access potential via the Web, we cannot help but be reminded of the rapidity with which it has been adopted into the expectations and acknowledgment of the general public.

Millions of content sources provided by individuals and groups are already accessible and countless more are becoming available on a daily basis. For educators, publications such as *Syllabus* and *T.H.E. Journal* focus on technology and education issues and regularly include information about Web sites housing pertinent information. Even governmental forces including the National Science Foundation, which has been active in funding numerous cyberspace efforts to improve education (Sattem), and Vice President Al Gore have significantly contributed to and impacted the Web revolution.

Alter's design for understanding IT (see Figure 1) clearly focuses the relationships between major components of the various driving forces influencing the Web's prolific expansion and influence on the business. The core, information technology, forms a key foundation of information systems which in turn represents a core component of business processes. These processes influence the firm and eventually the business environment (Alter). Of course, the interfaces are bidirectional.

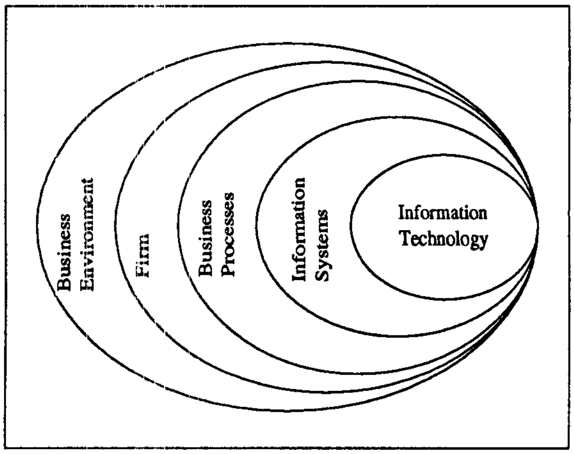


Figure 1

The major interactions exist between those closely coupled units such as IS and IT. However, we should also realize that the Internet which was once predominantly the domain of researchers and the government, has been efficacious adopted and intensely embraced by the business environment and the firm. It is now the responsibility of the IS professionals to quickly and seriously address the appropriate relationships between those intermediary factors such as business processes and information systems in light of this new application of technology to universal access method and means.

Some researchers, however, have shown that there is more hype than substance to the number of Web uses and thus true implementation of its purpose. Survey results reported by Derek Hulitzky, Director of Marketing for Computerworld's Professional Development Division and conducted by Computerworld, indicate that of the roughly 1.4 million IS professionals, less than 20% are Internet users, and less than 15% access the Web (Hulitzky).

However, the number of users in the public arena, from where the greatest demand shall be derived, appears to be continually increasing. Therefore, although professionals do not currently appear as enamored with it, the general public appears more prone to desiderate and request the image portrayed as universal connectivity regardless of its lack of universal substantive quality.

This predisposed demand points to the need for development of expertise in design and structure of both the technical and developmental components of Internet content and traffic. This, of course, implies revision and/or redesign of curricula and its components. Curriculum design, however, should be planned and directed based on an appropriate model. Regardless of the specific model used for developing IS curricula, needs assessment for any individual program involves five critical areas (Whitten):

* paradigm shifts
* industry directions
* student demographics
* community requirements
* facilities

Each of these needs can be directly connected back to the Web. Consequently, it is imperative that IS graduates are not only fluent in Web surfing as users, but can communicate effectively with regard to it, demonstrate proficiency in development, and become champions in its future progress. IS graduates will need to not only assess the viability, but also provide direction and control for offering appropriate services and linkages for corporations. Information systems will need to lead internal development, provide design parameters and models, and provide assistance to functional areas for development of this two-way interface with outside constituents. If the Internet becomes commercialized as many have predicted, the possibility of innumerable firms (thus job opportunities in IS) offering a variety of resources and value-added services is more than likely to appear on the horizon.

Motivation for Integration

Germane issues in IS education include declining enrollments, budget constraints, and the idea that industry may not believe that schools are indeed teaching what students need to know (Hensel) for success in the job market. These issues, in addition to the current impetus for using Total Quality Management principles, point to the need for curricula drivers to be in tune with the market and multiple constituents. As Elder points out, schools should be cognizant of their local market needs when designing and developing their courses (Elder). Additionally, he makes the point that traditionally delivered courses in the business schools often do not provide strength in the information systems areas. The Data Highway is highlighted as one of the ten key emerging technologies which must be recognized by college educators (Hoplin).

This is clearly evident when carefully examining academic disciplines and understanding how quickly Internet-accessed resources have become imbedded into curriculum content. Students in science and engineering regularly expect to download information directly from research centers and sites. From elementary schools engaging in on-line projects with students in other countries to university faculty collaborating with colleagues around the world, the communication provisioning of the Internet appears almost ubiquitous among educational institutions. The life cycles of projects in both academia and industry have been significantly reduced as collaborative work has gained freedom from time constraints that were traditionally imposed when participants were geographically dispersed (Mitchell).

While multiple academic areas have seen and begun to derive benefits from use of the Internet and Web related resources, few have accepted the responsibility of curriculum redesign for effective integration of the new technologies. This may in part be due to the inescapable difficulties associated with the dynamic nature of Web technology. In order to maintain currency, using resource material directly from the Web may supersede the use of a standard textbook, case book, or other traditional sources of information. Hence, the likelihood of non-IT related disciplines shouldering the burden of teaching about the Web and its appropriate management and implementation as either a business unit or tool is slim. It is therefore vital that IS immediately recognize the need and design methods and techniques for handling how, where, and in what form Web concepts should be disseminated in the information systems curriculum.

Method of Integration

Clearly, the integration of Web technologies into the IT curriculum has become an imperative and immediate need. Although the specific orientation of the faculty and the position of the IS department in the college or university will determine the focus of the teaching activity as well as its depth and breadth, but now is the time for IS academic planners to immediately assimilate the task of including Internet and Web concepts, tools and techniques into the IT curriculum. Integration may be accomplished by blending and interweaving Web concepts throughout an existing curriculum or specific Web importance may be established by introducing and including one or more relevant courses into existing curricula. Although the latter is the preferred route, given the great difficulty of new course development and integration, the former option may be more quickly and easily implemented.

Course work should be judiciously designed to include the essential concepts of Web browsing, page creation and maintenance, networking knowledge, programming knowledge of HTML (Berners-Lee), JAVA, and PERL, and multimedia application development. It is important to concentrate not only on the construction of Web applications, but also on design criteria and development guidelines. The exhaustive inclusion of these concepts would provide a functional baseline of knowledge to the WWW. Additionally, students may be able to develop some mastery at this level.

Web site management and advanced applications in multimedia will be highly dependent upon the available resources of the school or department. For example, JAVA programming which allows for seamless delivery of multimedia applications implies that end to end resources are available for viewing this level of application. Issues such as providing the appropriate blend of hardware and software paraphernalia for supporting this level of delivery can easily be placed in those provisioning sections of IS courses currently in existence. To provide greatest effectiveness for IS students, however, the experience of development and implementation of the multimedia applets is highly recommended.

Revising Existing Courses

Curricula currently encompassing a course in networks or telecommunications could quite easily adapt existing content to include WWW concepts.. It is, after all, the most widely known and accepted wide area network outside that of the telephone, or voice delivery systems. Good comparisons can be made between the highly regulated voice and traditional broadcast media with that of this unregulated, loosely controlled interconnected computing network. The use of the same physical media would be areas of common concept and design, but implementation and management would diverge significantly.

Access to and participation in the Web has become commonplace. One unique nuance of the Web is that we can use the Web to study the Web, much like we create C++ programs in order to understand the programming language. However, the difference lies in that while using the Web, we are not only viewing web contents, but also design and implementation of web resources. An analogy would be a C++ class using coding trails, program code, and outputs from all previous classes to help in the study of the language while developing new program applications.

Another approach may be to look at the business processes and client interactions as well as enhancement of external information sources available to the firm for places in which to infuse these concepts. For example, information systems design courses (or course segments) could encompass appropriate materials regarding design, implementation and use of the Web and Web tools. Or, these could be applied to the specific functional areas of marketing, management, production/operations, finance, etc. Cooperation with faculty from those disciplines may provide good support for changes in the course content for both IS and each of these functional areas.

Finally, Web usage may be promoted on a general departmental or school level. Class exercises and homework assignments may be structured to require resource search and retrieval from Web sites, instructor solutions may be availed solely from a class Web site, etc.

Regardless of the opportunities for inclusion in existing courses, IS professionals must use and design courses to include the pertinent concepts as indicated above. IS programs that are graduating students who lack the ability to design and build even an appropriate Web page will diminish the marketability of the graduates and once again disappoint many of the market constituents.

Creating A New Course

Prior to the creation of a new course, thorough analysis and design of course content relative to existing curriculum components is imperative. The paradigm shift is evident as even non-IS courses are implementing network aided instructional components. The industry and community demands will continue to drive a demand for Web-proficient IS professionals.

Programs may be elastic enough to allow for course components to fulfill the need for Web knowledge at appropriate levels. Programs which have developed such flexibility into their curriculum design, as suggested by Hoplin, will be more likely to react in a timely fashion (Hoplin) and smoothly integrate a newly developed Web-based IS course. For more structured programs, creating and integrating a new course may be a long and arduous task typically involving a requirement to gain approval via the appropriate academic channels. While it is not the purpose of this paper to debate the curriculum design problems and issues of academia, it is important to consider these parameters when upgrading IS curricula to allow for currency and adaptability. Despite the difficulties associated with curriculum enhancements, curricula visionaries must delve into the redesign and redevelopment process in order to ensure a robust academic program. The networks of the future will expand significantly and IS professionals will be their designers and champions.

Course construction for teaching Web concepts can occur fairly rapidly. Many Web sites containing course content and design information can be consulted. Additionally, Email, FTP, and listservs can be utilized for swift and productive sharing of ideas among colleagues in an ecumenical fashion with all interested parties or with selected individuals. A monopolistic dependence on industry and academic conferences as the exclusive forum for a healthy and timely exchange of ideas is no longer mandatory.

Although a singular "right" course of action or methodology for curriculum redesign to incorporate Web concepts has not yet gained prominence, several encouraging avenues are available. For example, a general course addressing Web concepts could be planned and offered as a core course in the business school curriculum. Although the schematics of the course would be drafted in the IS department, the course content could be chosen so as to dispense copious knowledge to all business majors. Michigan State University is using a $500,000 grant to develop curriculum materials that use the Internet (Meyers).

Another route is to offer a new interdisciplinary course in the IS curriculum which would entice multiple constituencies across a campus. The content of this course could include the use of the Internet as a research tool, various browsers (both commercial and non-commercial) and their usefulness as well as evaluation of quality information and sites for specific interdisciplinary resources. This could be established as a case-based course or as a capstone or partnership with library sciences or a business strategy and policy course. The University of Alaska, Fairbanks currently offers an MBA course organized around helping managers become intelligent collectors and consumers of on-line data (Lehman).

Another solution may be found in enhancing existing courses. Perhaps courses addressing networking technologies need to be redeveloped and redesigned such that augmentation of Web concepts can be accomplished without significant time delays. If one applies the design methodology proposed in (Goldman), the data communications and networking curriculum model applies a cognitive hierarchy to the conceptual architectures of local area networks, Internet works, and wide area networks. The fusion of the practical skills and conceptual architecture lends itself well to incorporation of either a specific course or modifications of multiple courses with Web concepts and technologies. This model also provides a strong design foundation for improving the quality of current networking courses.

Finally, if a curriculum affords the luxury of a Current Topics or Special Topics experimental course, the university demand and impact of a Web concepts course can be gauged without the outlay of any curriculum restructuring efforts. This course may initially be offered as an elective; if successful, the course can be incorporated as a mandatory segment of the IS curriculum.

Conclusions

Because information technology is "becoming increasingly integrated with business processes as companies gravitate toward employees versed in more than strictly technology and having more knowledge about how the company needs to operate" (Golden), institutions of higher education, responsible for educating and empowering the future workforce, must create opportunities for students to experience and appreciate the benefits of technological approaches to business solutions. Technologically versed employees, irrespective of their specific disciplines, reap the rewards of knowledge acquisition when participating in the job market. A recent US Current Population Survey indicates that workers who use computers on their job earn wages that are 10% to 15% higher than the wages of otherwise similar workers who do not use a computer at work (Krueger).

The information technologies explosion in our global society has created tremendous challenges and opportunities for educators to help educate and develop our future pioneers. The World Wide Web is the latest technology holding many promises and opportunities for progress and success. What will happen to the Web in the future is indeterminable at this point; however, its presence today and in the immediate future is indisputable. Rather than having diminishing importance, Web technology has rapidly been developing to provide increased functionality and hence continue to appeal to the masses as the communication medium of choice. This includes, for example, the usage of the Web not simply as a hyperlinked based GUI, but with the integration of JAVA applets technology, as a medium for sales transactions, database manipulation, etc. Additionally, the Internet and the Web provide the backbone for innovations in pedagogy including distance learning (Moskowitz), inter- and intra-university virtual classrooms (Goyal), and computer supported collaborative work.

REFERENCES

Alter, Steven, Proceedings Of The Tenth Information Systems Conference, Sponsored By DPMA Education Foundation, Phoenix, AZ, November 1993, p. 127.

Berners-Lee, T. (ed.), "HyperText Mark-up Language," CERN,http://info.cern.ch/hypertext/WWW/MarkUp/MarkUp.html.

Chapin, A. Lyman, "The State Of The Internet: 1995." Telecommunications. Jan 1995, Vol 29 No 1, p. 24-26.

Cohen, Lance. "Augmenting Instruction In Business Communication Courses With The Internet." Bulletin Of The Association For Business Communication Dec 1994, Vol 57 No 4, p. 31-35.

Conhaim, Wallys W., "The Internet: Linking Up To A Global Network." Link-Up. Jan/Feb, 1995, Vol 12 No 1, p. 5-7+.

Elder, Kevin L., "Implementing IS-95." Proceedings Of The Worldwide Information Systems Conference, Denver, Colorado, June 11-14, 1995, pp. 134-139.

Golden, Kathleen, "Resiliency Required," Insurance & Technology. May 1994, Vol 19 No 5, p. 36-40.

Goldman, James E., "Design And Implementation Of A Practical Business Oriented Undergraduate Data Communications Curriculum." Journal Of Information Systems Education, Summer 1994, Vol 6 No 2, pp. 80-85.

Goyal, Amita and Alka R. Harriger, "Using CSCW Technologies to Connect Classrooms," Journal of Information Systems Education, Volume 7, Number 2, Summer 1995, pp. 63-66.

Hensel, Mark. "The Approaching Crisis In Computing Education: Enrollment, Technology, Curricula, Standards And Their Impact On Educational Institutions." Proceedings Of The Eleventh Information Systems Education Conference, Louisville, Kentucky, October 28-30, 1994, pp. 56-63.

_____. "Hey, What's This Web Thing?" Datamation Jan 15, 1995, Vol 41 No 1, p. 38.

Hoplin, Herman P., "Integrating Emerging Technologies In It/is: Don't Let Your MIS Function Be A Malfunction Junction." Proceedings Of The Worldwide Information Systems Conference, Denver, Colorado, June 11-14, 1995, pp. 150-154.

Hulitzky, Derek E., "The Net, The Web, The Myths And The Realities (Don't Believe The Reach To "Millions")." Proceedings Of The Worldwide Information Systems Educators Conference, Denver, Colorado, June 11-14, 1995, pp. 38-43.

Krueger, Alan B, "How Computers Have Changed The Wage Structure: Evidence From Microdata, 1984-1989." Quarterly Journal Of Economics, Feb 1993, Vol 108 No 1, p. 33-60.

Lehman, Lisa M., "Driver's Ed For The Information Highway: A Required Course For MBA Students." Journal Of Information Systems Education, Spring 1995, Vol 7 No 1, pp. 30-33.

Meyers, Jason. "The Internet Hits The Books." Telephony. Oct 9, 1995, Vol 229 No 15, p. 28.

Mitchell, James E., "Engineering Education, Technology, and Change" Syllabus. March, 1996, Vol 9 No 6, pp. 38-40.

Moskowitz, Robert. "Distance Education." Link-up. Nov/Dec 1995, Vol 12 No 6, p. 38-40.

Sattem, Kathy. "NSF Tackles Ch.E Education Issues Online." Chemical Engineering. Jul 1995, Vol 102 No 7, p. 43.

Semich, J. William, "The World Wide Web: Internet Boomtown?" Datamation. Jan 15, 1995, Vol 41 No 1, p. 37-41.

Sutton, John. "The Management Archive." Business Communication Quarterly. Sep 1995, Vol 58 No 3, p. 40-41.

Whitten, Jeffrey L., "Development And Implementation Of An IS Curriculum." Workshop Conducted At Worldwide Information Systems Educators Conference, Denver, Colorado, June 12-15, 1994..

Teaching Hypertext Markup Language in the First Computing Course: One University's Experience

Jeanine Meyer
Pace University
Information Systems Department
1 Pace Plaza
New York, NY 10038 (212)-346-1569
meyer@pacevm.dac.pace.edu

Stuart A. Varden
Pace University Information Systems Department
861 Bedford Road
Pleasantville, NY 10570 (914)-773-3527
vardenf@pacevm.dac.pace.edu

ABSTRACT

This paper describes one university's experience in teaching Hypertext Markup Language (HTML) as an integral part of a required undergraduate course in computing. The choice of HTML was made because it incorporates in one facility several key themes of the course, including multimedia and hypertext, the Internet and the world of interactive on-line services offered through telecommunications facilities, team work, and organizational and presentation skills. Students are taught the elements of the language, are placed in teams, and create team Web documents. After a successful piloting of this innovation, the university now has introduced HTML into all sections -- totaling about 1,500 students per year -- of its undergraduate core curriculum computing requirement.

Introduction

The content and appropriate role of the first course in computing has long been debated by computer educators, and continues to be a topic of lively discussion today. Should it be a "terms and tools" course, or a course that focuses on the societal impact of the information technology, or the first step in laying down the foundations for formal computer studies, or something else entirely? The fundamental question is: What student outcomes in computing should be expected of all undergraduates upon graduation?

Many colleges and universities have sidestepped this question by not having a required computing course. Instead, an introductory course in computing is often found among a pool of science or "quantitative" electives from which a student is required to choose. In these institutions, many students are permitted to graduate without having any formal instruction in computing.

Pace University, however, includes in its undergraduate core curriculum a three credit computing requirement. The great majority of students, about 1,500 per year, satisfy this requirement by taking CIS 101 -- Introduction to Computing. The justification for maintaining the requirement in computing rests largely on the belief that people today and in the future will need to understand and effectively use information technology to be successful and productive citizens. Moreover, the course emphasizes the development of creative problem solving and teamwork skills through a number of group projects. An examination of social and ethical issues also forms an important part of the course.

Although the course is listed as three credit hours, it actually meets four hours per week, two in a large lecture format of up to seventy two (72) students, and two hours of lab with a maximum of twenty four students per lab section. Both lecture and lab sections are taught exclusively by professors or lecturers. Active discussion and structured group assignments are used to help students explore concepts and issues pertaining to the course, while the labs are hands-on skill-building sessions.

Overview of CIS 101

The lecture/discussion proceeds somewhat independently from the lab, though certain topics receive treatment in both places. A typical sequence of topics is:

1. Introduction; Computers are programmed.
2. Hardware and Software overview; Types of machines.
3. Computer Data; Binary arithmetic; Character strings.
4. Procedures; Flow charts; Serial versus concurrent processing.
5. System development.
6. Input and Output.
7. Storage concepts.
8. Languages, Subprograms; Operating Systems; Suites and Groupware.
9. Communications systems.
10. Information systems in business; Database management.
11. Multimedia; the Internet; and Industry Standards.
12. Computers in Society.
13. Review week: reflection on software used; Analysis of computer ads.

The thirteen lab sessions typically follow this sequence:

1. Introduction to personal computer and mainframe (used for E-mail) (1 week)
2. Structured Problem Solving and Programming Concepts Using QBASIC (5 weeks)
3. PC Productivity Tools Using Excel (3-4 weeks)
4. Multimedia Authoring and Group Project (3-4 weeks)

Although most of these topics are familiar ones for a first computing course, the three-to-four week section on multimedia authoring is a departure from the norm. Nevertheless, there are many good arguments for devoting a substantial portion of an introductory computing course to multimedia and hypertext.

Before discussing these advantages, let us first define the terms "multimedia" and "hypertext." The following definitions are based on the ones given by Sigle [1].

Multimedia is an orchestration of material composed of some, but not necessarily all, of the following elements: full motion video, still images, animation, computer graphics, text and audio. This material is generally delivered on a computer system.

Hypertext refers to the ability to create and browse through complex networks of linked documents. Unlike standard printed material, hypertext provides a way to navigate through documents in a non-linear, non-sequential manner. In its simplest form a user can choose a "hot word" on the screen and immediately get more information about that "hot word" (push down), and can easily return (pop back) to the previous screen. Hypertext does not appear by accident. It is the result of artful design and invention by the authors.

Using these definitions, it can be observed that multimedia and hypertext more closely model the way people think than traditional media. Most media (books, movies, television) are sequential and simplex. That is, there is a distinct beginning and end with an expected route to be taken, and all communication is one way. People, however, tend to think associatively, in which one idea leads to another in a largely unanticipated manner as one thought spontaneously triggers the next. Anyone who has wondered through the Wide World Web -- perhaps today's leading example of a hypertext system -- surely has experienced visiting a Web site that was totally unanticipated a few minutes earlier.

Secondly, there has been much discussion of the "MTV generation" and the effects of the media on the ability and willingness of today's students to concentrate on classroom work [2]. The importance of this effect is unclear, but there is no doubting the intrinsic appeal of multimedia.

If one accepts the value of multimedia and hypertext as an important component of an introductory computing course, then the next question that arises is: What system and platform should be used to develop these ideas? Pace has selected Hypertext Markup Language in a Windows 3.1 environment.

Why Hypertext Markup Language

Prior to the Spring, 1996 semester, the multimedia experience was provided mainly through the vehicle of LinkWay Live, an IBM product designed to develop multimedia presentations. It was used successfully for several semesters, and most students seemed to enjoy and benefit from the experience.

The sudden and rapid development of interest in the Internet and, in particular, the World Wide Web, over the past three years led Pace to consider the possibility of using Hypertext Markup Language (HTML) as a multimedia authoring language instead of a commercial system, such as LinkWay Live. The main arguments favoring the adoption of HTML are discussed below.

Multimedia and Hypertext Support

Although multimedia and hypertext have been in use in the classroom for many years [3,4], Hypertext Text Markup Language and the World Wide Web are relatively recent developments. HTML is the language used to build home pages or Web documents on the World Wide Web. Multimedia and hypertext are integral design features of the Web as originally conceived by Dr. Tim Berners-Lee at CERN, the European Particle Physics Laboratory in Geneva, Switzerland in 1989. Multimedia support available through the Web includes text, graphics, audio and video.

The Internet

No single technology phenomenon in recent memory has generated as much media attention, discussion, and high expectations as the Internet and the World Wide Web, and, for once, the reality may well live up to the hype. The use of HTML provides an excellent vehicle for learning about the Internet and discussing issues such as privacy rights, security, electronic commerce, the proper role of government in controlling access to information and infrastructure building, and the potential for "telecommuting."

Technical developments such as Integrated Services Digital Network (ISDN), Asynchronous Transfer Mode (ATM), and digital compression, are now on the verge of providing moderately priced, high bandwidth telecommunications facilities on a large scale. These facilities and their resulting products will soon offer a rich variety of high speed multimedia services directly to the homes and offices of subscribers. The pace at which this will become a reality has been recently accelerated by the passage of the Telecommunications Reform Act and the agreement between Visa and MasterCard on a common international security protocol.

The use of HTML and Internet also provides a good way to introduce the elements of the client/server model. Students learn that machines called "servers" are where information available through much of the Internet are stored, and that "clients" are machines through which end users gain access to the information through communications software and a browser. The Internet and its associated protocols provide the transport of messages between the client and server machines.

Organizational and Presentation Skills

In CIS 101 students are asked to develop a multimedia project using HTML. When preparing a multimedia presentation, the student designers must address the questions of how to organize information and to decide on the visual arrangement of text and graphics [5].

Graphics were produced using Paint Shop Pro. Since the LinkWay Live system had its own paint program, this was viewed as a significant increase in complexity for the system. However, students appear to be able to cope with it, though the amount of graphics is somewhat less.

The faculty have taken a variety of approaches to the choice of topic in both the old and the new multimedia module. One stand-by is to make a report on "The Computer Industry Today". The notion here is to make this a capstone assignment for the course. This assignment can be made more structured by requiring students to include material from product advertisements, job listings, and current event news stories. Another tactic is to ask students to find a news article or do research on the Web, and prepare their presentation on that topic. Many faculty allow students to choose their own topics, but get it approved by the second week of the module. Some faculty allow students total freedom.

Team Work

The ability to work with others effectively is a requirement for many, if not most future jobs for all our students. Certainly, employers have articulated this as a requirement for the information systems students. For this reason, the Information Systems faculty are beginning an initiative to build team work and presentation skills into our curriculum in a systematic way. Students will be expected to move through a progression of levels. The CIS 101 course serves as the first experience.

Faculty can take different approaches to team work. Speaking very generally, the teacher can assign the roles or allow the students to manage their own work. When students are grouped into different responsibilities, it is sometimes called the production company model [6].

One of the authors had the experience of coordinating a student "home page" development project. The class was divided into four areas of responsibility: (1) textual content authorship, (2) graphics, (3) formatting and hypertext design, and (4) HTML programming and testing. Each group, consisting of about six members, had a specific set of tasks that needed to be completed, while all the groups were required to coordinate their activities so that the total project could be successfully completed. Thus, there were both intragroup and intergroup skills that had to be addressed by each member of the class.

Allowing students to manage the assignment of work by themselves can be equally appropriate and sometimes brings quite gratifying results. The other author has had several occurrences of essentially the same circumstance: a subset of a team timidly approach the faculty and confess that Student A is a really skilled artist and is it okay that she or he do all the graphics for the project. They often go into great detail to demonstrate that they are doing more of the other work, such as research and writing. One reaction taken by faculty is to say that this is fine. However, mainly because the students seem to need something more, we may ask them to demonstrate that they can invoke the graphics program and draw a circle.

Wide Acceptance and Low Cost

A major practical advantage of using HTML is that it is easy to learn, widely used, and inexpensive. Once students understand the role and operation of a Web browser, they can be taught a simple subset of HTML "tags", say a dozen, which control the format in which textual and graphics material will be displayed on the "client" machine by the browser software.

In the case in which files containing HTML commands (tags) and graph files (GIFs) are maintained on a student's diskette, the browser can be instructed to load the files "locally." In this way, the student's computer can serve as both server and client machines. Thus, the Internet or any network whatever is not needed to create hypertext and multimedia presentations. In fact, local access is the typical way in which Web documents are developed and tested. The HTML files may be prepared using an HTML editor program, such as HTML Assistant Pro, or by using a simple word processor.

If the school has a server that is a host machine on the Internet, student Web documents can be stored there for general Internet access. Students can be provided with a vast collection of "clip art" or "audio clip" files, all downloadable for free from the Internet. Browsers and HTML editors also are available via the Internet free of change.

Finally, the general acceptance of HTML as a Web document preparation language provides students with a tangible skill that is widely recognized in the marketplace.

Implementing the Change

Pace University is a multi-campus institution, and CIS 101 is offered on three campuses spread out over a distance of about thirty five (35) miles. To ensure consistency between the campuses, a policy of a common course syllabus and texts has been adopted. Further, one of the co-authors is the University-wide coordinator for the CIS 101 course. All significant changes in the curriculum are implemented under her guidance after discussions with appropriate faculty members and committees. This centralized control has proved very effective in maintaining uniformity of course content across the many sections of the course that are offered each semester.

The use of HTML as a multimedia authoring tool was piloted in the Fall, 1995 semester. These "early adopters" reported that HTML was well received by students and many of the instructional objectives of the course were achieved through its use.

Training Instructors

Twelve (12) lecture sections and twenty nine (29) lab sections of the CIS 101 were offered in the Spring, 1996 semester. Some instructors were already comfortable with HTML. For those who were not, a half-day hands-on orientation session was conducted in January, 1996 where the basic ideas of HTML were covered. A small subset of the commands or "tags" was also introduced. One instructor who had taught one of the Fall semester pilot lab sections was on hand to share her experiences in teaching HTML to CIS 101 students. Instructors had little difficulty in mastering the basic ideas and skills of HTML.

Selected members of the University's Department of Academic Computing also attended the orientation, so that they would better understand the resources and services that would be needed to support students and instructors in using HTML.

Developing Instructional Support Materials

An advanced graduate student was assigned the task of developing a tutorial disk written in HTML to introduce the elements of the language. This excellently prepared tutorial was distributed at the instructor orientation along with a printed document on HTML. Other Web demonstration files were stored on a Novell LAN in the computer lab.

Upgrading Computing Facilities

The introduction of HTML into the CIS 101 course is paralleled by the University's commitment to upgrade computing and networking facilities throughout the University. The University is in the process of rewiring the campuses to provide Internet connectivity. This has already been accomplished at some of the computer labs, and a new fully equipped computer classroom, which seats thirty (30), was opened in Fall, 1995. The library also has been very active in providing Internet access through its "Electronic Learning Laboratory."

These improvements in facilities, however, will not be capable of handling the potentially thousands of students who will be comfortable and will want to use the Internet in coming semesters. Thus, the University is in the process of issuing a $43,000,000 referendum to support improvements in the University's physical plant and infrastructure. A sizable portion of these funds will be devoted to information technology enhancements, including networking and Internet access.

Early Observations

We consider the HTML training program a success because, even though it was described as an option, all faculty implemented production of HTML pages for the Spring, 1996 semester. We attribute this success to several factors:

1. As we have indicated earlier, the Internet and the World Wide Web is a well-publicized phenomenon and faculty want to be part of it. More specifically, we and the rest of the faculty want our students to be more than just consumers. The short unit will introduce them to the possibility of being creators as well as demonstrate the technical, organizational and aesthetic requirements.

2. Our approach of encouraging "early adopters" and then asking them to present the work of actual students performed within the time frame was effective. It allowed a period of limited use for Academic Computing and provided "existence proofs" for the rest of the faculty.

3. Many faculty were looking for something to replace LinkWay Live. This is an older DOS package, with some difficulties in implementation on the campus LANs. However, it had many excellent features, most importantly ease-of-use. Most students really enjoyed producing the multimedia folders and did excellent work. The production of HTML pages appears to offer nearly, though not quite, the ease of creation, with the additional, significant feature that the students can compare their work with work on the Web.

4. HTML production requires file management. That is, a typical project will involve multiple files, generally of different types and different locations, and students must keep track of the multiple parts. The faculty appreciate that we can include this challenge in a natural way as a culminating assignment. Students, for the most part, can handle it and thus have their understanding of various issues reinforced.

A small number of projects from the pilot program were selected to post on the University's Web server. We intend to make this an on-going activity. This means we have to confront the issue of how to choose from among a very large number of projects, with what standards of quality and taste. This has already emerged as an issue.

Concluding Thoughts

Many students come to the CIS 101 believing that what they need is training in a specific set of tools. Sadly, some faculty from other departments also believe this. Changing what we do demonstrates to students that new tools and technologies emerge all the time. We can also model for our students how we learn the new tools. The Web is, indeed, new but we, the faculty, can learn it by building on concepts and experiences that we have. In this way, students can move on from the class confident that they will be able to cope and thrive in the future.

References

[1] Sigle, John. "Multimedia and CAI Authoring Systems." **Proceedings in the Information Systems Education Conference**, Phoenix, AZ, 1993.

[2] Meyer, Jeanine. "Thoughts and Experiences on Multimedia Education." Technical Report, Number 92. School of Computer Science and Information Systems, Pace University, New York, February, 1996.

[3] Pearson, Rodney. "Hypergraphics: Technology-Based Classroom Presentations." **Proceedings of the Business Schools Computer Users Group**, Hershey, PA, 1989.

[4] Cummins, David E. & Ullrich, James R. "Using Hypercard in a First College Course in Computing." **Proceedings of the National Educational Computing Conference**, Boston, MA, 1989.

[5] Nanard, Jocelyne & Nanard, Marc. "Hypertext Design Environments and the Hypertext Design Process." **Communications of the ACM**, Number 38, 8, August, 1995.

[6] Agnew, Palmer, Kellerman, Anne, and Meyer, Jeanine. **Multimedia in the Classroom**. Allyn and Bacon: Boston, MA, 1996

Panel Session - Managing Emerging Technologies: Bridging the Gap between Academia and Industry

Dr. Kevin Lee Elder
Angelo State University
kevin.elder@mailserv.angelo.edu

Dr. Mark Hensel
University of Texas at Arlington
hensel@uta.edu

Dr. Herman P. Hoplin
Syracuse University
hphoplin@mailbox.syr.edu

This panel discussion will center around the difficult task of managing emerging technologies and the challenges IS professionals encounter in Industry and Academia.

Major items coming into focus include managing IS automation, IS innovations, and the Impacts of IS on the organization and the IS professional's new role. A future look at applications development will expose trends and practices which will have an enormous impact on both Industry and Academia.

As a result of the dissatisfaction with traditional development methods, three trends are emerging that could have major implications for the future of applications development. First, some types of software are becoming disposable. That is happening because organizations demand faster development of their applications. The implication is that many organizations are willing to sacrifice quality for speed in software development because the applications developed today become obsolete in a very short time.

Second, rapid applications development (RAD), object-oriented development, and network-centric computing seem to be emerging as the development methods of choice in many organizations. The implication is that although structure is still needed to properly analyze and specify a development project, speed of development is essential.

Third, languages such as Sun's "Java", Bell Labs' "Inferno", and others may make it possible to gather "applets" of code from sites on the Internet and use them in new, rapidly developed applications. Browsers could become the "Windows" environment of the Web. The implications are that application development may become speedier, and that the training for a career in the development field may change dramatically.

The trends in application development mean that the field, the discipline of applications development is changing again, and changing very quickly. The IS industry, IS educational institutions, and IS professionals must change just as quickly to meet the demands of the future.

The IS Model Curriculum must address these emerging technologies and their impact. We discuss these emerging technologies and expose students to the skills necessary to utilize them, but we don't use them to teach the information systems curriculum. We need to be using the emerging technologies in our classroom instruction to prove how valuable they are to the industry our students will graduate into.

If faculty took more time to develop lectures they would not use the same examples to teach the old material. Dr. Andy Whinston in his keynote address to the Information System Education Conference, Louisville, 1994, addressed the need for IS faculty to rethink how we convey our knowledge to students. With the advances made on the Internet, the Worldwide Web, digital libraries, cd-rom storage devices and hypermedia we are in the dark ages when we use only a textbook, overheads, and the chalkboard to convey knowledge.

Emerging technologies could in fact help bridge the gap between what industry expects and what academia is currently providing industry. A partnership between Industry and Academia should be established to foster this plan for managing the emerging technologies.

The Development of Proficiency Exams to Measure Computer Literacy

Larry W. Cornwell and Matthew K. McGowan
Business Management and Administration
Bradley University
Peoria, IL 61625
309-677-2314 and 309-677-2279
larryc@bradley.bradley.edu and mmcgowan@bradley.bradley.edu

Abstract

Computer literacy has been a requirement in most American Assembly of Collegiate Schools of Business (AACSB) undergraduate programs. Incoming freshmen are now entering these programs with a level of competency that suggests they do not need to take the traditional computer literacy course. Students need to be familiar with the University's computer environment to complete assignments effectively in their business programs. This paper describes the development and implementation of proficiency exams to measure students' computer literacy in an AACSB program. The staffing and scheduling of proficiency exams and seminars in proficiency areas are also discussed.

Introduction

This paper describes an approach taken to establish a level of competency in computer literacy for entering students at Bradley University's Foster College of Business Administration, FCBA. Entering freshmen are now coming to the University with a wide range of computer experiences. The old requirement, making all students take the same introductory course, is very difficult to teach with the broad range of student backgrounds. The required course may not be a valid requirement for students who are competent at some of the topics. The purpose of the old course was to introduce the student to the computing environment of the University.[1] Some students can very easily adapt to this computing environment without taking a course. On the other end of the spectrum, some students prefer or need to take a course to develop the level of competency needed.

In an effort to address this problem and address the new AACSB requirements[2], the Foster College of Business Administration decided to change the requirement of a computer literacy course at the freshmen level. The old requirement was a three credit hour course that consisted of a lecture part for two hours per week and a hands-on laboratory part for one hour per week. The lecture part covered business information systems concepts and the laboratory part covered productivity software in the University's computing environment. The new requirement has moved the coverage of the business information systems concepts to a three hour course at the junior level, BMA 372. The laboratory part is taught in a one hour course but the FCBA students cannot use the credit for this course toward FCBA total hours required for graduation. Students may satisfy the FCBA computer proficiency requirement by passing a competency exam.

Description of FCBA Computer Requirement

The FCBA Curriculum Development Committee presented the following description and rationale for the new requirements:

"*COMPUTER PROFICIENCY (0 semester hour) AND BMA 172 BUSINESS COMPUTER SKILLS AND APPLICAITONS (1 semester hour) Students must demonstrate proficiency in various categories of computer skills. Topics covered will include word processing, E-mail, Internet, file transfer, DOS, spreadsheet, database, statistical package, business graphics, and presentation graphics. (This list is not fixed nor is it necessarily exhaustive.) The demonstration of proficiency must be completed by the time the student has accumulated 24 semester hours of credit at Bradley.*

This proposal is prompted by a number of considerations. First, the emphasis is clearly on the demonstration of computer competencies which we highlight in our college objectives. Second, the focus on assessment of skills is consistent with the approach we must adopt under the new AACSB assessment criteria.

Third, and perhaps most important, students are entering the university with an ever-expanding breadth of computer skills. Therefore, the demonstration of skill proficiency, rather than a prescribed course requirement, seems to permit curricular flexibility, based on the level of student development. Further, this stance represents a set of expectations and assumptions that are likely consistent with the skill base of our future students.

Regarding the demonstration of computer proficiency, students will be self-paced in this process. If necessary, students may acquire additional background and skills necessary to pass the proficiency tests by participating in training modules that will be offered at regular intervals throughout the academic year. However, some students will enter with sufficient background to complete all proficiency categories with little additional training. Others may need to avail themselves of most, if not all, training modules. Importantly, the gap between skill competency and entering skill levels is realistically growing smaller. Further, most students possess skill levels of sufficient content that can acquire the necessary skills in the self-paced environment that is proposed.

Indeed, in the short-run, some students may enter with no background or such low levels of skills that the self-paced training modules will be ineffective. These students may elect to take the proposed 1-semester-hour BMA 172 to provide the necessary background and training. This hour should be treated as a remedial hour of credit. One may logically assume that the number of students needing, and electing, to take this remedial option will grow smaller as the level of pre-college computer exposure continues to grow. Eventually, the 1-hour course option may be unnecessary.

Please keep in mind that computer skills are included as one of our "across the curriculum" categories. Therefore, all courses should provide the inclusion of relevant computer applications.

***BMA 372 (3 semester hours) INFORMATION TECHNOLOGY** The need for a 3-hour course may be prompted by a variety of considerations, but 3 specific factors weigh heavily on this proposal. First, while all of us recognize that our fields of study are growing and expanding, the area of information systems is literally exploding. We must equip our students with the fundamental awareness and understanding of the technological advances in information and communication systems, and a sensitivity to how these technologies will affect the way businesses operate. Logically, this leads to a second consideration. This course will emphasize data access and use for competitive strategic purposes. Indeed, the strategic use of information for business decision-making must be a strong focus of any progressive curriculum. Third, the course must include an interdisciplinary focus so students can see how various disciplines may apply course themes. This may be facilitated through cases or projects."*

Makeup of the Proficiency Exam

The proficiency exam is designed to establish a level of competency in using the computing environment at Bradley University. The topics that were taught in the original introductory course comprise of three areas: (1) Communications, (2) Spreadsheet applications, and (3) Database applications. This distribution of topics is listed below:

Test Area	Topics
Communications	Word-processing E-mail Internet FTP
Spreadsheet Applications	Creating Worksheets Graphics
Database Applications	Creating Databases Creating Reports/Labels Querying Databases

We decided that the best way of measuring competencies would be in the form of a hands-on proficiency exam. The proficiency exam is made up of three tests that measure the level of competency in each area.

The communications exam requires the student to prepare a multi-page document in a word processor on a topic that can be researched on the Internet. Instructions specify the topic, resource web sites, and special format for the document. Once the document has been prepared, the student transfers the document to a second computer. The student is then asked to attach the document to an email message that must be sent to the grader.

The spreadsheet applications exam requires the student to develop a spreadsheet for a business problem. Instructions specify the problem, basic data, assumptions, needed calculations and special format for the spreadsheet. The student is asked to answer several what-if questions and to produce several graphs.

The database applications exam requires the student to create a database structure. Data is provided in a spreadsheet format that needs to be imported into the new database structure. The student is then asked to query the database and produce answers to the queries in a report format.

The proficiency exam approach is expected to ensure that our students are more uniformly prepared for the microcomputer applications course. Students enter the microcomputer applications course with a variety of backgrounds. Transfer students frequently receive credit for an introductory computer course, but they may not have experience in the environment (communications package, spreadsheet package, and database software) which is assumed for the microcomputer applications course. These students will be able to take the module(s) needed to gain familiarly with our environment.

Seniors frequently elect to take the microcomputer applications course in the hopes of gaining a competitive edge over other students entering the job market. Under the present system, these students may have had the introductory course during their freshmen year, possibly four or more years before taking the microcomputer applications course. These students will be able to "brush up" on software that they have not used extensively during their subsequent years. In some cases one or more of the software packages used for the microcomputer applications course is different from the package taught in the introductory course. Students can learn the new software under the modular approach.

Competency Lists

The competency requirements are designed to represent a level of proficiency needed for a student to effectively use computers in FCBA courses. Students must demonstrate competence in each of the topical areas: communications, spreadsheet applications, and database applications. It is also an assumption that all students have a familiarity with basic DOS and Windows operations.

The communications area includes using word processing, a skill which most freshmen possess before attending college. An increasing number of students will have been exposed to electronic mail and the Internet, since use of these technologies is increasing in secondary education. There is still a need to be able to perform these tasks in the FCBA computing environment.

1. <u>Communications Competency List</u>

Word Processing

- Create a document
- Save document in ASCII format
- Use basic features
  Font: Underline, Bold, Italics, Super/Subscript
  Headers, footers, Page numbers, Spell check
  Import a document from another format

E-mail

- Send and receive mail in our computing environment
- Become familiar with an email system

FTP

- Perform basic file operations on host (UNIX) system: list files, change directory, create directory
- Upload and download files in ASCII and some other format

Internet

- Access and navigate gopher menu system in our environment
- Perform basic navigation in WWW environment

The spreadsheet application area reflects the need for students to be able to create basic models for decision making. Students need to be able to create formulas in their models and to present the data in various formats.

2. <u>Spreadsheet Application Competency List</u>

- Create a spreadsheet
- Use basic formatting functions
  Font: Bold, Italics, Underline
  Number formats: percent, fixed, etc.
- Create multi-worksheets (pages)
- Create and use formulas
  Arithmetic operations
  @functions: sum, max, min, avg
  Absolute and relative addressing
- Perform Editing functions
  Copy, Cut, Delete, Paste cell
  Insert, Delete row, column
- Create simple graphs

The database application area exposes students to the requirements of structuring data for reporting. Students are required to complete queries on a data file which is large enough to make it difficult to answer the query by inspection. This affirms the importance of using computers as a tool for manipulating data.

3. <u>Database Application Competency List</u>

- Create a database structure
  Use field name, type, width
- Perform basic editing
  Add a record, modify a record, delete a record
- Perform basic queries using a QBE type of query tool

- Create and run reports
  Include titles, totals
- Import data from ASCII, spreadsheet, or another database format

Grading of Proficiency Exams

The proficiency exam is administered by a graduate assistant who uses a checklist to evaluate the student's exam. The graduate assistant marks the checklist to indicate the characteristics of the student's exam that match the checklist items. The checklist ensures a level of consistency in the grading process. The point total awarded for each exam is compared to the cutoff score needed to pass the exam. Since there are three proficiency exams (one for each area), all exams that are passed will be recorded. The student will only have to repeat the proficiency exams that were not passed. When the student passes all proficiency exams, the FCBA official responsible for maintaining student records is notified that the student has passed the proficiency requirement for the College.

Alternatives for Satisfying Computer Proficiency Requirement

Students entering FCBA can satisfy the computer proficiency requirement in several ways. The alternatives are listed below:

1. Pass the one hour course, BMA 172 The one hour course, Business Computer Skills and Applications (BMA 172), may be completed. The course meets 1 day a week for a semester. Although the student earns one hour credit for the course, it does not count toward graduation in the FCBA. The course is offered on a pass/fail basis and passing the course will satisfy the FCBA computer proficiency requirement.

2. Pass the proficiency exam A student must pass the proficiency exam in each of the three areas before completing 24 semester hours. Failure to complete the proficiency exam in this time limit requires that the student enroll in BMA 172. In an effort to help the student prepare for the proficiency exams, review seminars are offered over the topics covered for each exam. A full complement of seminars for a specific exam is offered during the week preceding the exam.

Typical Offerings of Proficiency Exams and Seminars

1. Offerings of Proficiency Exams During a semester, the group of three proficiency exams are offered four times. Each group (communication, spreadsheet applications, and database applications) is offered every month. The three exams are scheduled during a given month and repeated every month during the semester. One group of three exams is offered at the end of the semester. This provides four opportunities for the student to complete the proficiency exams. In the fall semester for example, the three exams are scheduled in September, October and November. The group of three exams is also scheduled the first week of December.

2. Offerings of the Seminars To help a student prepare for a specific exam, seminars are offered covering the topics of that exam. The seminars are similar to the one hour modules covered in the BMA 172 course. The following seminars are scheduled to cover the proficiency areas:

Area	**Seminar**
Communications	DOS-Windows Email-FTP Word Processing Internet-WWW
Spreadsheet Applications	Spreadsheet Basics Spreadsheet Basic 2 Spreadsheet Graphics
Database Applications	Database Structures Database Reports Database Queries

Staffing

The staffing of the administration and implementation of the proficiency exam requirement is performed by one faculty member and three graduate assistants. The faculty member is responsible for creating the scripts for the modules used in the BMA 172 course and the seminars. Another responsibility is the creation of the proficiency exams and the checklists for grading the exams. The responsibilities of the graduate assistants vary. One graduate assistant has the responsibility of teaching four sections of BMA 172. Each section meets once a week for one hour and fifteen minutes. Other responsibilities include grading of homework, grading exams, conducting help sessions and conducting office hours. The second graduate assistant has the responsibility of teaching seminars throughout the semester. The seminars are scheduled to cover the topics needed for the next scheduled proficiency exam. During the semester, thirty seminars are scheduled. Another responsibility of the graduate assistant is to help students identify the seminars and other materials needed to prepare for a specific proficiency exam. The third graduate assistant has the responsibility of administering the proficiency exams. A full complement of the exams (communications, spreadsheet applications, database applications) is scheduled for every month. The graduate assistant must also maintain records of exams passed and report to the FCBA official the list of successful students. An additional responsibility for the second and third graduate assistants is the maintenance of additional materials (hardcopy and electronic) which can help a student prepare for a particular topic on the exam. Both graduate assistants maintain office hours where students can meet with them and identify materials that can help improve their computer skills.

Alternative Seminars: Future Considerations

In considering the objective of the computer proficiency requirement in the FCBA, alternative seminars are a natural development. It is not reasonable to expect an entering freshman to have skills in using statistical, presentation, or project management software. It is reasonable to expect the use of these types of software packages in the FCBA curriculum. If seminars are scheduled for SPSS for Windows, PowerPoint, and MS Project, faculty can announce the seminar schedule to the students and request that the students attend the seminars. This will remove the need to use valuable class time for software training and aid the integration of computer usage in the FCBA curriculum.

Summary

Business students need to be familiar with the University's computing environment in order to complete assignments in various courses. Instructors of upper level business courses assume students are computer literate with respect to the University's computing environment, and students need to be computer literate when entering the job market. The approach described here ensures that incoming students are computer literate in the FCBA's computer environment. This approach requires students to pass proficiency exams in each of three topical areas: communications, spreadsheet applications, and database applications.

We have developed competency requirements for each topical area. We developed a checklist for each proficiency exam based on the competency requirements, which also ensures that the exams are consistently graded. Students can demonstrate proficiency in one of two ways: pass the proficiency exams or pass a one semester hour course that covers the three topical areas.

Students who have a deficiency in one or more topical area can take seminars in the areas that they are deficient. The seminars are offered several times per semester, and the proficiency exams are scheduled to coincide with the completion of the seminars. The seminar approach permits the possible addition (or deletion) of topical area seminars as necessary to familiarize students with the tools needed to complete course assignments successfully. The seminar style is also conducive to "refresher" modules taken by students who need to reacquaint themselves with topical areas. This approach recognizes the variety in levels of computer literacy among incoming students while still ensuring students are literate in the FCBA computing environment.

---

[1] Cornwell, L., Fairfield-Copple, A., Rubash, K., & Stephens, P. (1995). " A Model for the Computer Literacy Course, "*ISECON '95 Proceedings: Meeting the Challenge of a Global Marketplace.* " Education Foundation of the DPMA, Charlotte, NC, November 1995.

[2] AACSB (The American Assembly of Collegiate Schools of Business). (1991). *Final report: The AACSB accreditation project,* March 1991.

Computer Literacy: Teaching it in the Late 1990's

Bret R. Ellis, Ph.D.
ellisb@byuh.edu
Edward Jensen, Ph.D.
jensene@byuh.edu
Elizabeth Cluff, MS
cluffe@byuh.edu
Kylie Barker, student
Natchiya Nakkhongkham, student

Information Systems Department
School of Business
Brigham Young University-Hawaii
55-220 Kulanui Street
Laie, HI 96762
808-293-3472

Abstract

Five sections of "Information Technology (DPMA 95-1)" classes were taught using traditional lecture or computer assisted learning and evaluation tools. This paper provides some useful findings for reducing the overhead of teaching general education computer literacy courses, suggests that students perform as well or better using computer assisted instruction and evaluation, and demonstrates how these computerized tools can provide effective instruction in a wide variety of ethnic backgrounds now prevalent on college campuses.

Background

According to the DPMA 95 model curriculum, the purpose of this course, Information Technology (95.1), is to provide a broad understanding of computer concepts and software applications. Traditionally the teaching of the "Information Technology" course has been taught through lecture. Today, at many universities there is the shortage of faculty. This university shortage causes courses that are less important to the department major to be relegated to part-time, adjunct, or student teachers. A typical less important course includes any general education course such as "Information Technology." In addition, the evaluation of the students understanding of the course material is done through regular written quizzes and exams. Often this type of evaluation provides limited feedback to the student as far what they do or do not understand. They simply receive a 79 on the exam and move on to the next section or chapter.

In todays MTV world, students expect fun and exciting learning. And like the 24 hour stores of today, they also expect that they can get the instruction when and where they want it.

In the "Information Technology" course, the desired type of learning or in the words of Gagne, et. al, the expected learning outcome from this type of course is called "Verbal Information." (Gagne, 1988) Gagne, et. al., has demonstrated that specific desired learning outcomes such as "Verbal Information" can be produced with specific classroom pedagogies. Pedagogies such lecture, small group instruction and individual instruction have been studied extensively. However, the effectiveness of multimedia in the classroom to supplement or replacement to the classroom lecture is still not completely understood. Some researchers and writers, however, suggest that multimedia and technology should be considered as alternatives.

Tapscott suggests that in the 1990's, education has been transformed using Information Technology. In "The Digital Economy" he writes "The results are spectacular...computers have reached the point at which they can help youth and others learn—these students feel empowered. (Don Tapscott, 1996)

According to Dr. Robert Blalock, Director of Learning Technologies Research for AMR Training and Consulting Group, computer based training embraces the following fundamental elements of effective learning:

1) Students are required to give active responses to the learning material.
2) Feedback from the computer is both frequent and immediate.

3) Quality computer-based instruction is self-directed and incorporates lots of practice.
4) Quality computer-based instruction is fun...and is an intrinsic motivator of achievement and enhanced self-esteem.

In a dissertation written by Hughes entitled "The Effectiveness of Multimedia Technology in the Acquisition of Spanish Vocabulary," verbal information is shown to be taught effectively using multimedia technology.

Research Questions

To better understand these issues the following four questions were used in this research:

1) What is the difference in overall grades between traditional course instruction and computer assisted learning and evaluation tool instruction in the Information Technology course?
2) Are students more efficient in study, classroom learning and quiz/exam taking using computer assisted learning and evaluation? Do they like the computer assisted learning more than traditional course instruction?
3) What is the students preference to learning computer concepts: traditional instruction (lecture) or computer assisted learning and evaluation?
4) Does the change in instructional style have an impact on the learning outcome of students from different ethnic backgrounds?

Procedures for the Study

Five sections of Information Systems 190 (IS190) "Introduction to Information Technology" were taught. Two sections were taught traditionally and were designated as the control group. These two sections met once a week in front of a member of our Information Systems (IS) faculty for lecture and discussion on that weeks reading assignment. Course quizzes and exams were proctored in the campus testing center.

The other three sections of IS190, known as the experimental group, were taught using computer assisted learning and evaluation tools exclusively. Both control and experimental groups used an identical text, review questions, quiz questions, and exam questions. The experimental group received two class periods of instruction by a faculty member in the use of the computer and the use of the instruction and evaluation tool. Thereafter, students went to the computer lab after reading a chapter to review the chapter material and to take the quizzes and exams.

Students in the control group met once a week in the classroom with the instructor for a lecture on the assigned chapter for the week. Students in the experimental group were required to take at least one quiz per week but were not restricted as to how quickly they could move through the chapters. For example, a student could complete and was encouraged to complete the course as quickly as their schedule and desire would allow.

When using the computer, the student in the experimental group had the following options when using the computer assisted learning and evaluation tool called WorldTutor: Access the demo (reviews, quizzes, and exams), the quiz (quizzes for chapters 1-14), and the midterms/final.

Following the completion of the chapter reading, students in the experimental group were encouraged to study the reviews until a competency was achieved. For example, a student was asked to pass at least twelve of twenty review questions that consisted of true/false, multiple choice, short answer and fill in the blank questions. If the student could not pass twelve questions, they were instructed to review the material again within the chapter until they could successfully pass the reviews.

The students in the experimental group received e-mail from the instructor to keep in touch with students, to assist with problems and to answer question about any difficult textbook content.

Findings

A total of 212 students participated in and completed this course and research project. There were no noticeable drop-out from any of the sections that could be attributed to the research project.

The first research question was "What is the difference in overall grades between traditional course instruction and computer assisted learning and evaluation tool instruction?" As shown in Table 1, which shows the overall grades of the students in the control and experimental group, the average grade for the control group was 2.39. The average grade for students in the experimental group was 2.59.

The second research question was "Are students more efficient in study, classroom learning and quiz/exam taking using computer assisted learning and evaluation?" "Do they like the computer assisted learning more than traditional course instruction?"

In the control group, the students completed the quizzes, the midterm and the final at the same time according to the syllabus schedule. In the experimental group, four students completed the entire course prior to the midterm of the control group. Many more of the students completed the entire course prior final exam date of the control group.

Control Group			Experimental Group		
		Number of		Number of	
Grade		Students	Score	Students	Score
A	4	8	32	17	68
A -	3.66	7	25.62	11	40.26
B+	3.33	14	46.62	9	29.97
B	3	6	18	28	84
B -	2.66	10	26.6	13	34.58
C+	2.33	12	27.96	8	18.64
C	2	8	16	12	24
C -	1.66	4	6.64	5	8.3
D+	1.33	5	6.65	5	6.65
D	1	3	3	5	5
D -	0.5	3	1.5	3	1.5
F	0	8	0	8	0
Total		**88**	**211**	**124**	**321**
Average Grade			**2.39**		**2.59**

Table 1

In Figure 1, the grouped responses from the students (experimental group) course evaluations are shown.

The students also liked the fact that no classes or paperwork were required in the class. They are thought the computer based reviews and evaluation helped prepare for the quizzes.

The third research question was "What is the students preference to learning computer concepts: traditional instruction (lecture) or computer assisted learning and evaluation?"

This question was not be adequately answered using the data received from the students.

The fourth research question was "Does the change in instructional style have an impact on the learning outcome of students from different ethnic backgrounds?"

As mentioned previously, this university has strong representation from various parts of the world including the Pacific Islands and Asia. The university categorizes the target population is five categories: U.S., Hawaii, Pacific Islands, Asia, and Other.

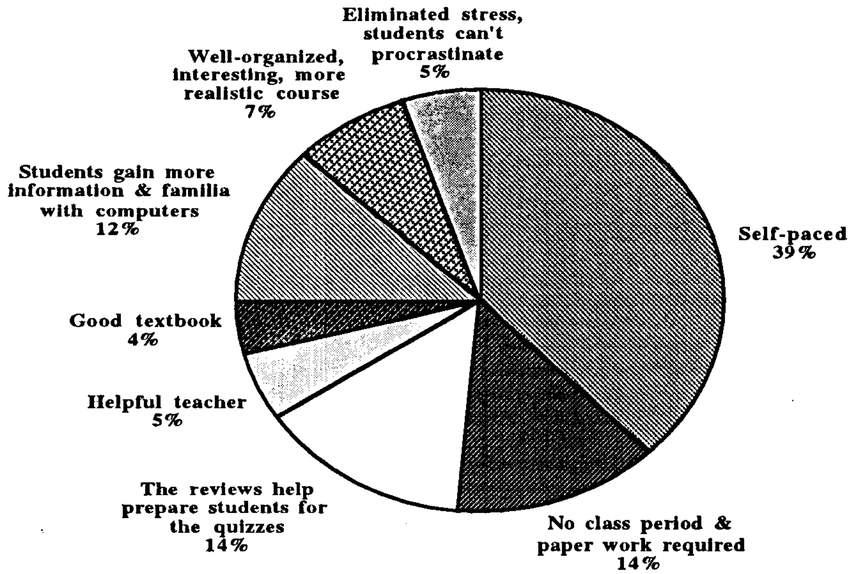


Figure 1

Students from the U.S. and Hawaii performed nearly the same in both the control and experimental groups as shown in Figure 2. Students from the Pacific Islands scored twice as high in the experimental group. Asian students and students from other countries in the experimental course scored almost twice as high.

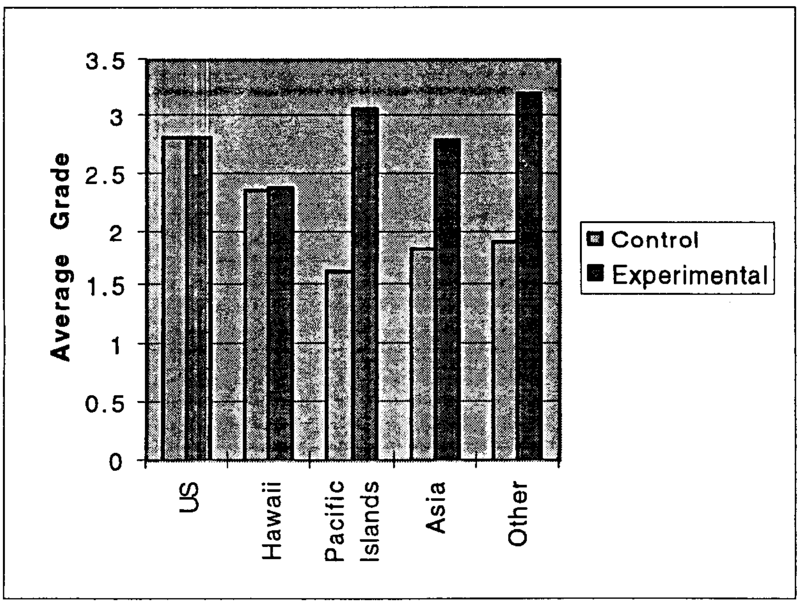


Figure 2

Based on the course evaluations shown in Figure 3, it can be noted that environmental factors and control are still a concern. Providing sufficient computers and ensuring that students do their own work is still a problem. In addition, we noted that students doing reviews and taking quizzes/exams had to contend with typical lab distraction and noise.

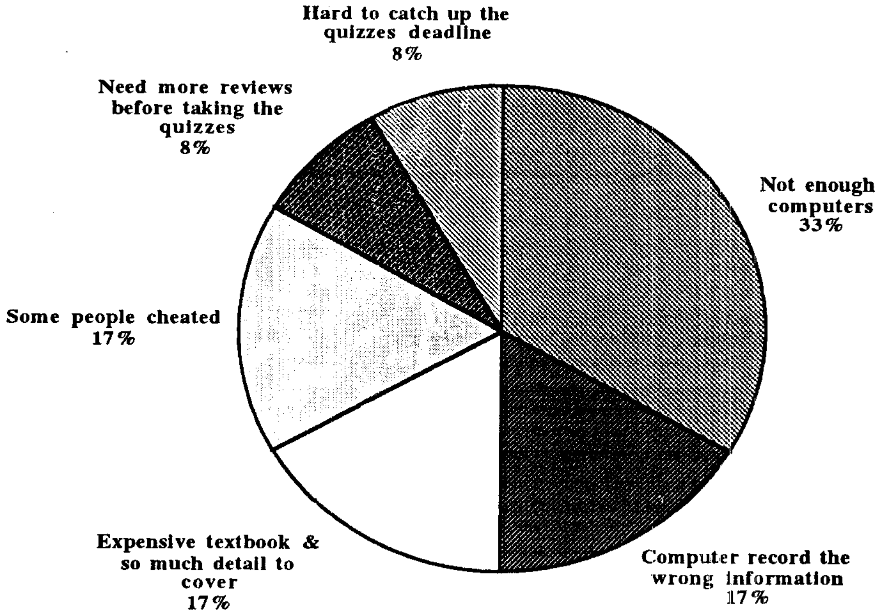


Figure 3

Conclusions and Recommendations

Based upon the original four research questions and the findings of the research here are the conclusions and recommendations:

1) Students in the experimental sections of the “Information Technology” course did as well and better than those in the control sections. In can be concluded then that the beginning course in Information Systems can be taught using computer assisted learning and evaluation tools and it can be expected that the student will gain at least the same amount of knowledge.
2) Students in the experimental group completed the quizzes and exams earlier in the semester than the control group. The students appeared to prefer this type of instruction over the traditional lecture format. Based on previous study, it should be expected that students will do better in this type of course because they are in control of their time and the pace of the course.
3) The data for the third research question “What is the students preference to learning computer concepts: traditional instruction (lecture) or computer assisted learning and evaluation?” was not sufficient to make any conclusions or recommendations.
4) Perhaps the most important finding and conclusion of this study was that students that are often at a disadvantage because of deficient language skills perform as well or better than there English counterpart in this course when using computer assisted instruction and evaluation tools.

Future Research Questions

Based upon the findings, conclusions, and recommendations, the following research questions should be pursued to further understand the implications and applications of computer assisted instruction and evaluation tools in the classroom.

1) Could this type of course be extended to other lecture type introductory courses with the same success?
2) What is the relationship between a students score and the amount of time they take to prepare for the quizzes/exams?
3) What is the relationship between students that read the chapter, utilize the on-line review, and then take quizzes versus students that read the chapter and then take quizzes only.

Summary

Using research related to how people learn, a course was developed that provides both review, instruction and evaluation via the computer. The student preferred this type of instruction for this type of "verbal information" course. Students from all different ethnic backgrounds can learn and perform well using this type of course.

References

Blalock, Robert H., "Effective Adult Learning." AMR Training and Consulting Group White Paper, 1993, p 3.

Gagne, Robert M., Leslie J. Briggs, and Walter W. Wager, "Principles of Instructional Design." Third Edition, Holt Rinehart Winston, 1988.

Hughes, T. F., "The Effectiveness of Multimedia Technology in the Acquisition of Spanish Vocabulary." University of Pennsylvania Dissertation, 1993.

Tapscott, Don., "The Digital Economy." McGraw-Hill, 1996.

A Shared "CORE" Curriculum For Information Systems (IS), Software Engineering (SE) and Computer Science (CS) Based on a 1995 National Survey

Herbert E. Longenecker, Jr.[1], David L. Feinstein[1], John T. Gorgone[2], Gordon B. Davis[4], J. Daniel Couger[3], Ron L. Williams[1]

[1]University of South Alabama, [2]Bentley College, [3]University of Colorado at Colorado Springs, [4]University of Minnesota

Introduction

During the past several decades significant effort has been expended in developing curricula in the computing disciplines. As the professions have matured, there has been a diversification of programs. While there may indeed be a clear separation of the professional computing disciplines (Glass 1992), it is the premise of this paper that there is considerable overlap in the body of knowledge relative to three of these programs, information systems(IS), software engineering(SE) and computer science(CS). Furtherore, despite relatively different mission statements, we believe that many of the exit objectives that faculty hold for their graduates will be relatively similar. Even if the exit objectives are not completely aligned, it will be possible to find considerable common ground during the first two years of study. Therefore, it should be possible to develop a core curriculum suitable to the diverse programs. There is considerable evidence to support this premise. A survey conducted for the development of the IS'95 model (Longenecker 1995) suggested that there is considerable interest in forming cooperative linkages between CS, IS and SE programs. Within the Software Engineering Institute curricula(Ford 1990,1991), a dependence of SE on the CS curriculum model (Turner 1991) is recognized. Within this paper we present data obtained during the IS'95 review process that can be used to support the concept of a common core program.

Consider the following definitions of the three disciplines:

> **Computer Science** can be defined as "*the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation, and application. The fundamental question underlying all of computing is, what can be (efficiently) automated?" (Denning 1988)*
>
> **Information Systems** involves "*... complex socio-technical entities that have taken on critical roles in local, national and global organizations*" which "*provide support for the goals of the organization and its management -- strategic, tactical and operational -- in a timely and cost effective manner*" "*to improve the performance of people through the use of information technology...where the ultimate objective is performance improvement...where the focus is the people who makeup the organization...*" (IS'95; Sprague 1993, p14)
>
> **Software Engineering** is "*...the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software..." (IEEE 1990). "The two concerns that pervade software engineering are the complex requirements of systems and the need to build them economically in a for-profit environment. The context of software engineering tends to be software intensive systems that have substantial performance (real-time), capacity, reliability, security, and safety requirements; the discipline addresses how such systems are built and maintained in ways that are economically viable for the producers and users." (Ford 1995)*

In this paper we ask the following questions:

1. Is it possible to construct a mission statement for a core sequence that can be aligned with the missions of the three disciplines of IS, SE, and CS?, and then,

2. Given such a mission, can a curriculum system be identified that would produce students who can continue effectively in the respective disciplines?

Relationships among Body of Knowledge Elements

Since there are established bodies of knowledge for the disciplines of CS, IS, and SE, significant relationships between these bodies of knowledge would imply the possibility of a shared core. IS'95 contains a complete body of computing knowledge. It consists of three major subdivisions: 1) Information Technology, 2) Organizations and Management, and 3) Theory and Development of Systems, which expand hierarchically to four levels under these primary areas. In total there are 510 elements.

The IS'90 body of knowledge (Longenecker and Feinstein

1991) was the basis for the IS'95 body of computing knowledge. It contains 244 total elements defining the IS body of knowledge. The IS body of knowledge was reorganized by the IS'95 task force, and the revised elements were the basis for the IS'95 elements.

The computer science body of knowledge (Turner 1991) is divided into nine subject areas, for which there are approximately 211 "topics" at the third or leaf node level of the hierarchy. To study relationships between the CS and IS bodies of knowledge we observed that given elements:

1) mapped 1:1 - matched an existing element, or
2) mapped 1:n - were contained within an existing element.

In the first case, 24 elements of the CS topics mapped 1:1 to third level elements of IS'95. In the second case, 187 CS topics mapped 1:n with IS'95 third level elements. While there might have been other possible relationships, these were the only two that were found. The effect of this mapping gives considerable richness through the added depth at the fourth level hierarchical expansion. An additional observation is that the CS topics were mapped largely to knowledge elements within the Information Technology area of IS'95.The software engineering body of knowledge (BCS 1989; Ford 1990, 1991) was condensed into 120 elements. Sixty nine (69) of these elements map 1:1 to the third level of IS'95, while 51 map 1:n giving a fourth level of expansion. This occurred primarily in the areas of software and systems development contained in the IS'95 Area 3, Theory and Development of Systems.

The software engineering curriculum suggests much of CS as a good foundation for undergraduate and graduate level software engineering degree programs. The SE curriculum contains specific CS knowledge elements which are compatible with CS 1991. Thus, the software engineering curriculum can be viewed as primarily consisting of two areas from the IS'95 body of knowledge, specifically, information technology and theory and development of systems. Figure 1 below shows the relative distribution of the knowledge elements.

----Computing Body of Knowledge----				
IS'95 Primary Knowledge Areas	Element Counts			
	Total Number	Information Technology	Organization and Management Concepts	Theory and Development of Systems
IS Body of Knowledge	244	72	86	86
CS'91 Body of Knowledge	211	166	19	26
SE'91 Body of Knowledge	120	40	16	64
IS'95 Total Elements	510	246	108	156

Figure 1 - Distribution of CS, IS and SE Elements Within the IS'95 Body of Computing Knowledge.

Achieving Exit Curriculum Objectives

Since curricula are a system consisting of related components (Gagne 1988), then sequences of courses must produce the desired system output. Thus, a First sequence (F) would be followed by an advanced sequence A as follows:

F ------> A

For any given program F would imply mission directed activities which are designed to cover a given body of knowledge in such a way that the goals are achieved. The question of finding a common core sequence for three related curricula would involve redefining the First (F) sequences for each discipline by enlarging the mission and goals of the core to be compatible with each of the disciplines to be served by the new central component. In addition, the advanced (A) sequences of the disciplines might have to be modified (MA) based on the derived core sequencein order to achieve the goals of the advanced sequence:

PROPOSED	EXISTING POSSIBLE ALTERNATIVE
$F_{CS} \dashrightarrow A_{CS}$	$\vert\dashrightarrow MA_{CS}$
$F_{IS} \dashrightarrow A_{IS}$	CORE -----> $\vert\dashrightarrow MA_{IS}$
$F_{SE} \dashrightarrow A_{SE}$	$\vert\dashrightarrow MA_{SE}$
	CORE = Union $\{F_{CS}, F_{IS}, F_{SE}\}$

Core Mission

To help in the definition of the common core, we propose the following core mission statement. This statement is compatible with the observation that these related disciplines have a strong requirement to develop skills in information technology and problem solving (Denning 1988, 1992). In addition, in keeping with IS and SE missions (Leventhal 1987; BCS 1989; Mulder 1993; Denning 1994; Longenecker 1995), software development should include the theory and application of systems concepts. These ideas lead to the following mission:

The mission of the Computing Core Sequence (CCS) is to provide a rigorous and complete treatment of those concepts and applications common to computer science, information systems and software engineering. The material in this core should form the bases for students to successfully continue in specially differentiated programs in CS, IS or SE. The students should explore a wide class of ideas relative to the computing professions including the information technology, organization and management concepts, particularly as applied to the theory and development of systems and software. This core must impart in the students enthusiasm for the computing disciplines and confidence in their ability to become successful practitioners. Students must have learned team skills and learned how to learn necessary skills to solve problems which arise and must be solved in realistic project situations.

A Starting Point for Courses

Since SE largely defers to CS programs for its initial experiences, the courses associated with CS and the proposed IS'95 model are incorporated in the table below.

We suggest that the recommendations for the first courses from CS 1991 be augmented in two ways. First, they should include the systems and IS concepts from IS'95. Second the formal concepts of systems development from both IS and SE should be added. Exposure to personal productivity tools such as spread sheets, database management systems, as well as code and application generators could be mixed with conventional 3rd and 4th GL solutions. Concepts of knowledge work and information could be explored. Learning a programming language (preferably an object oriented language) is a requirement. There has been considerable recent discussion regarding breadth rather than the more traditional depth first approach in the first course. The approach taken in IS'95 favored the breadth first development. This would be consistent with one of the models from CS'91. Exposure to concepts of operating systems and computer communications should be included within the first year sequence. These suggestions are shown in Figure 2.

We suggest that the IS'95 sequence (IS'95.4,5&6) be used as the basis for the second year sequence. This material should be augmented with SE concepts and the CS focus on Data/File/Database and Object Structures. Problems should be chosen from a general topic mix characteristic of all three disciplines. The IS'95 operating systems and telecommunications courses must cover both the practical and theoretic aspects of the topics. Figure 2 show both the existing and proposed core sequences.

Conclusion

It is possible to conceive of a common core sequence that is consistent with recommendations for curricula in CS, IS and SE. Specific elements from the common body of knowledge should be mapped to learning units (Longenecker 1995) to fully define the common model. Mission and philosophy statements for the courses and sub course units have to be aligned with the more general concepts expressed within the core sequence mission.

References

ACM 1983. "ACM Recommendations for Information Systems, Volume II", ACM Committee on Computer Curricula of ACM Education Board, ACM, New York, New York.

Ashenhurst, R. L.(Ed) 1972. "Curriculum Recommendations for Graduate Professional Programs in Information Systems", in ACM Committee on Computer Curricula of ACM Education Board. "ACM Recommendations for Information Systems, Volume II", ACM, New York, New York, 1983.

Briggs, Leslie J., 1977. "Instructional Design", Educational Technology Publications, Englewood Cliffs, New Jersey.

BCS 1989. "A Report on Undergraduate Curricula for Software Engineering", The British Computer Society and The Institution of Electrical Engineers, June 1989.

Existing and Proposed Core Course Sequences	
Existing Disciplinary Course Sequences	Proposed CORE Courses Supporting CS,IS,SE
Year 1	Year 1
Computer Science (2 semesters) Computer I,II **Software Engineering** (2 semesters) Computer I,II **Information Systems** (2 semesters) Fundamentals of IS Personal Productivity with IS	**Proposed CS/IS/SE Core 1** (2 semesters) Systems Theory Problem Solving Concepts and Practice Solution Identification and Development Personal Productivity with IS Tools Systems and Software Development Life Cycle Concepts Algorithmic Development Programming Information Technology Examples from CS Breadth First: Computer Architecture, OS, Telecommunications
Year 2	Year 2
Computer Science (4 semesters) Operating Systems Data Structures Computer Architecture Programming Languages **Information Systems** (3 semesters) Information Technology Hardware and Software Telecommunications Programming, Data and Object Structures **Software Engineering** (5 semesters) Computer Systems 1,2,3 Software Architecture 1 Software Analysis 1	**Proposed CS/IS/SE Core 2A** (2 semester courses) Programming, Data, File, Database and Object Structures: Data and Object Representations and Structures, Problems, ADT's, SE Methodology, Language Structures, Third, Fourth and Fifth GL Usage; GUI's, OO, and Event Driven Programming Client/Server Programming Project Management and Implementation Teams, Personal/Interpersonal Development **Proposed CS/IS/SE Core 2B** (2 semester courses) Computing and Communications Machinery: Architecture, and Operating Systems Archictecture, Windows and User Interfaces, Client/Server Configurations Telecommunications, Installation and Applications Communication Devices, Media and Networks Distributed /Client Server Implementations Network Configuration and Management OS Configuration and Management

Figure 2 - Comparison of Existing Courses and a proposed Common Core sequence of Courses.

Denning, Peter, 1992. "Educating a New Engineer", Communications of the ACM, Volume 35, Number 12, December 1992, pp 82-97.

Denning, Peter J. 1994. "Designing a Discipline of Software Design (Keynote Address)", Software Engineering Education: 7th SEI CSEE Conference, Jorge L. Diaz-Herrera, ed., Springer-Verlag, New York, New York, 1994

DPMA 1981. "DPMA Model Curriculum, 1981", Published by DPMA, Park Ridge, IL.

DPMA 1986. "DPMA Model Curriculum, 1986", Published by DPMA, Park Ridge, IL.

Ford, Gary 1990. "1990 SEI Report on Undergraduate Software Engineering Education", Technical Report SEI-90-TR-3, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., March 1990.

Ford, Gary. 1991. "1991 SEI Report on Graduate Software Engineering Education", Technical Report SEI-91-TR-2, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., April 1991.

Gagne, Robert M., Leslie J. Briggs and Walter W. Wager, 1988. "Principles of Instructional Design, (3rd Edition)", Holt, Rinehart, and Winston, New York, New York.

Glass, Robert L. 1992. "A Comparative Analysis of the Topic Areas of Computer Science, Software Engineering and Information Systems", Journal of Systems Software, (19:4) pp277-289.

IEEE 1980. "Draft Report on MSE-80: A Graduate Program in Software Engineering", IEEE Software Engineering Subcommittee of the Computing Society Education Committee, 1980.

Leventhal, Laura M. and Barbee T. Mynatt 1987. "Components of Typical Undergraduate Software Engineering Courses: Results from a Survey", IEEE Transactions on Software Engineering, Volume SE-13, Number 11, November 1987, pp 1193-1198.

Longenecker, Herbert E. Jr. and David L. Feinstein (eds.) 1991. "IS'90: The DPMA Model Curriculum for Information Systems for 4 Year Undergraduates", published by Data Processing Management Association, Park Ridge, 1991.

Longenecker, Herbert E. Jr., David L. Feinstein, J. Daniel Couger, Gordon B. Davis, and John T. Gorgone 1995. "Information Systems '95: A Summary of the Collaborative IS Curriculum Specification of the Joint DPMA, ACM, AIS Task Force", Journal of Information Systems Education, Volume 6 No. 4, pp 174-187.

Mulder, Michael 1993. " A Framework for Academic Programs in Informatics: Task Force Report", NSF Informatics Task Force Center For Advanced Computer Studies, University of Southwestern Louisiana, Lafayett.

Nagin, Paul and John Impagliazzo 1995. "Computer Science: A Breadth First Aproach with C", Wiley, New York.

Nunamaker, Jay F., J. Daniel Couger and Gordon B. Davis 1982. "Information Systems Curriculum Recommendations for the 80s: Undergraduate and Graduate Programs", Communications of the ACM, Volume 25, Number 11, November 1982, pp 781-805.

Tucker, Allen B. and Bruce H. Barnes (Eds.) 1990. "Computing Curricula 1990: Report of the Joint Curriculum Task Force (draft document)

Turner, A. Joe, and Allen Tucker, (eds) 1991. "Computing Curricula 1991 - Report of the ACM/IEEE-CS Joint Curriculum Task Force", Communications of the ACM, volume 34, number 6, pp 68-84, July 1991.

Educational Uses of Common Gateway Interface Programming

T.M. Rajkumar
Decision Sciences and MIS
Miami University
Oxford OH-45056
Email: rajkumar@miamiu.muohio.edu

Abstract:

While HTML allows documents to be distributed, common gateway interface (CGI) allows the instructor to interact with the student on the Web, using forms. This paper provides an introduction to CGI, and explores the educational possibilities.

1 Introduction

The rapid growth of the internet has led to a proliferation of new educational uses of the technology. An area that has received significant attention has been the use of the world wide web (WWW) and hypertext markup languages (HTML). Many professors have established home pages using HTML, and use the home page as the starting or jump off point for the students in their courses.

The use of home pages has obvious benefits for many course related instruction, such as distribution of syllabus, assignments, and class lectures/notes to the students. This enables the instructor to make any last minute changes to the content of the course, and have the student be aware of changes to the lecture/instructions prior to coming to class. Many home pages of instructors, allow the student to send mail easily to the professor via the home page using the browser. This allows students who are not familiar with the school's mail system to contact their professors easily. Home pages also have indexes to sites of relevant interest to the course. This gives additional research resources to students that they may not have been aware earlier. These resources can be linked to specific projects/papers in the course to provide additional viewpoints.

The above advantages are not meant to be exhaustive but a sampling of what is possible with home page design. Many instructors spend considerable amount of time making their home pages visually appealing, intuitive to navigate and useful to the student.

All the above types of uses take advantage of the ability of the WWW to display HTML documents. HTML by itself allows the user only to mark up documents, display graphics, forms, tables etc. HTML does not allow the instructor to interact in a more dynamic manner with the student. Education involves not only distributing documents through HTML, but also allowing the instructor to interact with the student in other ways such as providing feedback, giving suggestions, taking part in discussions etc. through the Web.

A mechanism that allows such possibilities is the common gateway interface (CGI) built into the WWW specification. The aim of this paper is to introduce CGI, its educational uses, and concentrates specifically on interacting to databases on the Web. The next section introduces CGI, and means to interact with it using the language Perl. This is followed by the educational uses of CGI. Section 4, introduces the database applications with CGI, its problems and considerations. The last section concludes with a look at new and emerging technologies such as Java.

2 What is CGI

The WWW is a distributed client server system, with the web server distributing the document that the browser (client) requests using a specific uniform resource locator (URL). The web server can extend the functionality of a web site using CGI programs (Bobrowski, 96).

CGI is the standard for communications between a web server and a server-side gateway program. When you access a gateway program, the server executes the gateway program and passes it any data that the client may provide. Once the gateway program has executed , it sends the result back to the server which then sends it back to the client. (Graham, 96 p. 419). This process is depicted in Figure 1. CGI specifies how data is passed from the server to the gateway program and vice-versa.

Gateway programs can be written in any language such as Visual Basic, C, C++ or as executable scripts using languages such as DCL, Apple Script, Perl etc. Scripting languages such as Perl are more common since the scripts

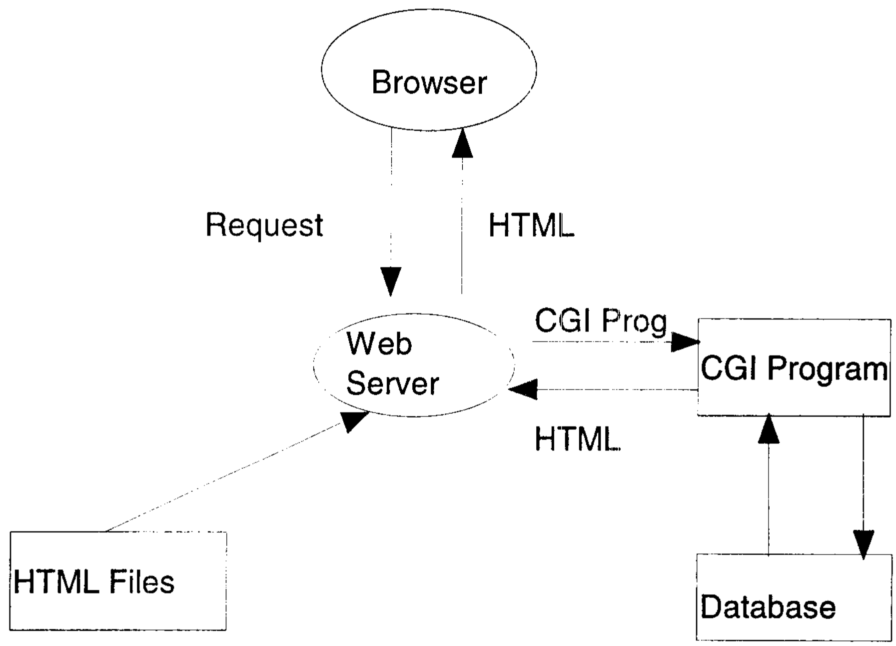


Figure 1 CGI

are portable and easy to write. In addition, extensive libraries and modules exist to make CGI programming easy for the user. The speed of execution of the script gateway program is in general not the limiting factor but the resource the gateway connects to. Hence, languages such as Perl are used extensively in CGI programming.

A call to CGI program is typically encoded as a link to a HTML document. Clicking on the link causes the CGI program to execute on the server, by sending a form to the user. User data is then returned from the client to the server as data in a message body using the POST method. When a server receives the data from the client, it passes the data to the designated gateway programs standard input. When the gateway has completed processing the data, it sends the data as standard output. This is read by the server program, which then sends it back to the client. Since the data that the server receives from the gateway is sent to the client, the gateway typically formats the data as HTML statements. Since, the data that CGI programs can send can contain text, graphics or both it may be essential to include meta-information such as the Content type so that the client browser can display the information (Morris, 95). While standard input and output is the main means of exchanging data between the server and the gateway program, they also can exchange data by setting up special environment variables.

Perl is a scripting language that is widely used on the web to do CGI programming. A specific reason is the availability of modules that make CGI programming easy. One such module is the CGI.pm class (simple common gateway interface class). This library makes it easy to generate Web fill-out forms, and parse their contents (Stein, 95). It enables you to define CGI objects that contain the value of the query string. Using the CGI objects methods, keywords and parameters can be examined easily and create additional forms as necessary.

3 Uses of CGI

CGI programs can be put to a variety of uses. A simple use is to generate feedback from the user regarding a certain class, or a mid-term feedback etc. This is done through the use of forms. In general a form data is displayed on the client. Once the student has entered the data, the data is typically mailed back to the instructor. Figure 2, displays a standard form the author uses to get mid-term feedback. An advantage of this method is that it is totally anonymous. All accounts on the student systems simply generate WWW@miavx1 id. The specific student cannot be identified. It also enables students who missed class on a certain day to provide the feedback the next day.

A second use of forms is to take feedback at the end of a class day for the student. A simple on-line quiz is given (multiple choice), and the gateway program simply inputs data from each student into a file. This file can then be processed to get a tally of how many students understood the concept. This processing can be done by a second perl program that the instructor can run. This gives the instructor instantaneous feedback on whether students understood a specific concept or not. This technique

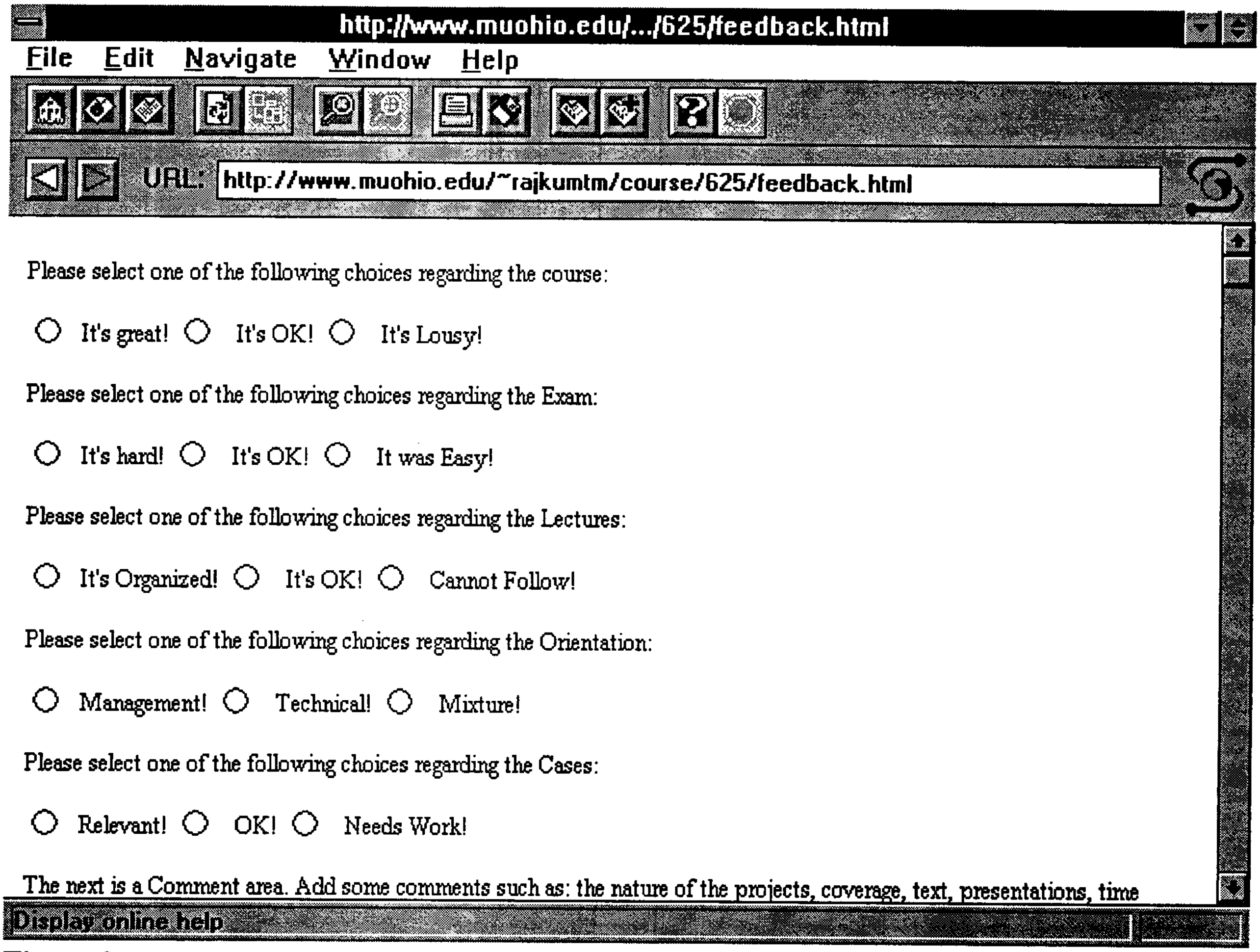


Figure 2 Feedback Form

assumes that the material is presented in a lab. An alternative mechanism is to encourage each student to take this quiz before a certain time on the next day of class, so that you can go over material that is not understood by a majority of students.

A third use of forms and CGI programming is to allow the student to upload an assignment file to a specified location on the computer. The student would instruct what file needs to be uploaded, and this can be copied to the server computer. This allows the instructor to not have to carry disks to and from the class. While this can be done through the web, it may also be possible to do it using the local area networks' capabilities and facilities.

A fourth use the author uses CGI and forms is to give grades back to the student. Currently the author uses a flat file (tab delimited fields) to store the grade on the server. The student enters their social security number, and submits the form (Figure 3). The program searches the flat file, and returns the grade, along with any comments that are needed (Figure 4). This enables the students to find out their grade earlier than when the exams are returned in the class. The exam answers are also put out on the web. Under the current design that is used, each exam, and project are kept in separate files, and the student has to enter their social security number multiple times. Alternate mechanisms to do it better are being studied including keeping it in a database, and trying some of the solutions discussed in the next section. There is also a program that allows the student to find the cumulative score, and the averages, median and high for the class, so that they can find where they stand in the class at any point in time. This allows the student to be always informed where he or she is with regard to the rest of the class.

4 Database Applications of CGI

Some problems with the approach used in the previous section to providing grades is the fact that it comes out of flat files. This leads to the following problems (Johnson, 95):

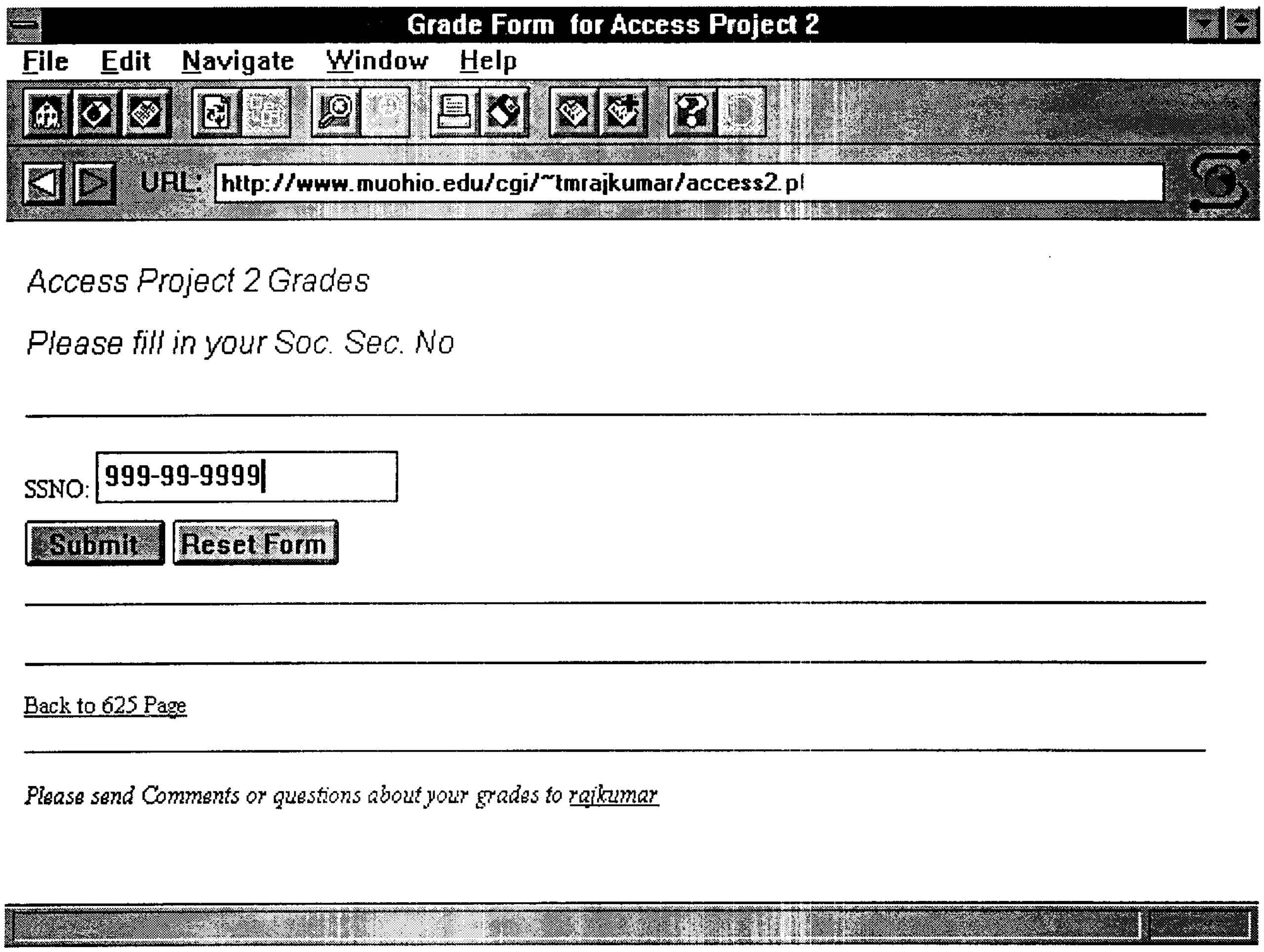


Figure 3 Grade Request Form

a) the data takes longer to retrieve
b) cannot maintain relations between the files automatically.
c) each query on the server spawns a separate process, that makes it highly inefficient.
d) and difficult to manipulate if the user wants to modify the data once the user has retrieved it.

These problems arise because of the inherent disparity between the web and databases (Dieckman, 95). The web is essentially a stateless system with a simple client server request/response model. The browser client, opens a communication to the server, issues the request, receives the results and closes the connection. Every query request results in a new connection which is terminated upon completion of the request, typically a time out period.

In contrast, databases are environments where multiple transactions occur in a single database session, with the state of the session being maintained between transactions. Between sessions, the database remembers the users' previous actions in the form of data stored in the tables (Haverty, 96). When ever a student enters some data,that information can be fed back to the student the next time around.

There are two basic methods to solving the above problem:
a) store and pass state information between client and server, so that each subsequent database session is provided with the information about the preceding history. State information can be passed back to the client by CGI scripts (typically as hidden fields in a form). Hence, subsequent requests from the user would return these fields back to the script, which can then parse the result.

b) An alternative approach is to have the CGI script maintain a database session and keep it open while processing subsequent client requests. Again identifying information is passed between the client and the web server each time so that it uses the proper database session. As users are wont to walk away without properly terminating, the CGI script must also terminate the

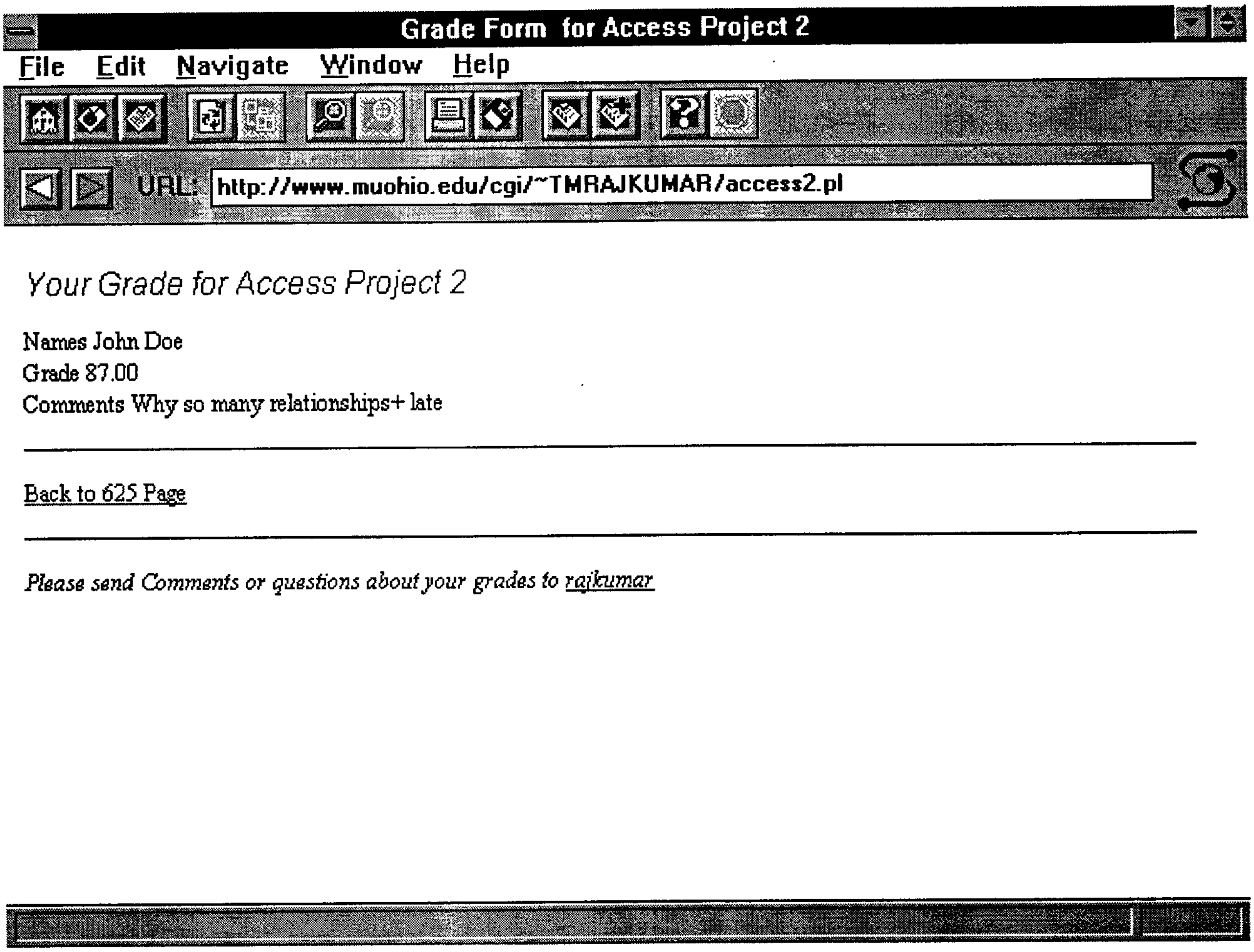


Figure 4 Grade Return Form

corresponding database session after a certain period of inactivity on the part of the user.

Gateways to link standard databases such as Oracle, Sybase, or front ends to SQL are available from the vendors or from the web itself. In general, these call a database backend program to process the query through SQL statements. Microsoft is also developing an application programming interface that allows users running PERL to access database information directly.

Another approach that is emerging is the use of gateway to link to an open database connectivity server (ODBC) to the web, as opposed to a direct database server. This gives us the advantage that the data can be in any type of database format and data can be retrieved as long as there is an ODBC driver. This provides the flexibility for the system to directly access corporate wide data by going through the internet.

Other approaches try to bypass the CGI overhead. For example, WebSql the product from Sybase allows embedded SQL to be processed directly (Frank, 95). Here, WebSql monitors all access to the server. Whenever embedded SQL exists, it preprocesses the page by performing the database query and merges the results into the page.

Security issues also abound in using databases on the net. Some aspects include (Dieckman, 95):
a) limiting access to the database
b) Choice of hardware and software
c) making more of the CGI portion of the script work within the confines of the secure database environment.

4. Emerging Solutions

CGI programs normally extend the power of the web browser by executing server-side programs. The emerging generation of web browsers allow the users to execute client-side programming features allowing simple distributed HTML applications (Bobrowski, 96). For example you might want the user to start a teleconferencing session using a link. This implies that the server might have to send information to the client requesting the execution of a program, or actually send a program to the client for execution. For example, the teleconferencing program could actually start a program and also pass the information needed to contact the other party.

The features supported by the new class of browsers include the following:
1. Execution of client-side scripts with a scripting language. This script is then executed in the client, and updates the form. For example, the students score on exam1, exam2, and the project grades may be given in a single form, and a client side script could evaluate the current grade for the student. In general, these scripts stay on the server and are downloaded prior to execution.

2. Downloading of compiled binary programs from the server that execute on the client. (A typical example would be an application developed using the Java development language). These scripts are used to provide more advanced client side processing. For example, the student may be taking an on-line exam, and the system may be responding with his score as soon as they enter the choice for every question and even graph their progress. Such constant updates, cannot easily be done in the previous approach, but is feasible with the execution of compiled binary programs.

Client side execution has tremendous security risks associated with them (Graham, 96). In general, client programs are run from within a "Secure shell". All executions start from a local shell that has knowledge of the programs that it can execute.

The second option is to design special client programs, and load it on every client PC, and allow the server to access only those predetermined safe clients. Programs exist that will allow the registration of these programs to the server. Java is an example of a language that allows the application to execute in the client computer and allows it to perform the above task easily by harnessing the power of the client machine while maintaining security. Java is a platform independent object oriented language from Sun Microsystems. Java combines the power of C++ but reduces its complexity, and supports memory management and dynamic binding, and has the advantage of having security built into it. For example, Java programs can access only a single directory on the users' disk, has public key encryption, and operates on object types, verifies the downloaded byte code and works within the operating systems security..

6 Conclusion

Technology on the internet continues to evolve at a very rapid pace. However access to databases and performing simple gateway programs are becoming standardized, as the web browser becomes the more common interface for many users. New uses such as registering for classes through the web continue to evolve. Educators need to keep up with this new technology and use it as yet another tool in their arsenal to deliver instruction and receive feedback to meet the educational needs of the student.

References

Bobrowski, S. "How to Use Web Technology For Building Client Server Applications" Oracle Magazine, Mar-Apr 1996, pp. 89-93.

Diekman, M. "When Worlds Collide: The Internet meets the Web," Database Programming and Design, Aug. 95, pp. 48-54.

Frank, M. "Database and the Internet." DBMS, Dec. 1995 pp. 44-52.

Graham, I. "The HTML Source Book", Addison Wesley, 1996.

Haverty, J. "How to Build Web Applications Using Oracle 7 and Oracle Web Server", Oracle Magazine, Jan-Feb 1996, pp. 86-94.

Johnson, J.T. "Web Servers get Ready of Some Real Work", Data Communications, Nov. 21, 1995.

Morris, M. "HTML for Fun and Profit", Sun Soft Press, 1995.

Stein, L. "CGI.LIb", http://www-genome.wi.mit.edu/ftp/pub/software/WWW/cgi_docs.html, 1995.

Using Groupware To Facilitate Teaching and Learning in MIS

Liang Chee Wee
Department of Economics and Business
Luther College
Decorah, IA 52101

ABSTRACT: The purpose of this paper is to share how a newly built electronic collaborative work facility has provided a window of opportunity to the management information systems (MIS) faculty, of a small, liberal arts college in the Mid-west, to examine the use of groupware creatively in MIS courses. Besides giving students a hands-on exposure to a groupware, the technology has enhanced active learning, communications, and overall participation in class.

1. The Forces of Change

In his 1993 article, Thomas Sandman identified seven principal forces affecting the MIS curriculum in his institution. They are: (1) the academy, which represents the body of knowledge and guidelines endorsed by the Association for Computing Machinery and the Data Processing Management Association; (2) the organizational constraints such as budgetary limitations; (3) the quality of the MIS faculty; (4) the impact of new technology; (5) the industry that hires our graduates; (6) the quality of MIS students; and (7) the competition posed by other institutions. The fourth element has influenced both the teaching and learning of MIS at our college.

1.1. The Round Table Room

In order for a face-to-face, verbal group discussion to be productive and meaningful, all the facts and relevant matters must be submitted before the group for discussion. Most of the time, however, the group members do not feel "free" to share their true feelings about certain matters. Maybe it is because the "true" feeling might be perceived as an attack on a group member. The situation is compounded when the group member concerned has more seniority than the person making the comment. In addition, only one person can talk at a time, and someone might dominate the discussion. Also, some others might be too timid to participate verbally at all. All these factors contribute to "process losses" (Nunamaker et al., 1991).

The Round Table Room (RTR) is an electronic collaborative work facility equipped with 25 networked computers and one facilitator station. The outer circle has 17 computers and the inner circle has 9 computers. Comments are submitted electronically and anonymously by all participants at the same time. The software license was obtained from the Ventana Corporation, developers of GroupSystems™ for Windows.

Since the meetings in the RTR are "a blend of face-to-face and electronic exchange of information, the room was designed to place the participants in as much of a circular pattern as possible." Furthermore, the RTR represents "the movement from rigid hierarchical relationships within a group or organization toward more collaborative and responsible sharing of responsibility and work, but within a carefully organized and designed atmosphere focused on producing effective group results" (Royksund, 1995).

The RTR technology grew out of group support systems (GSS) research at the University of Arizona. The first GSS room was constructed in 1984. It was designed to facilitate group processes such as generating ideas, organizing ideas, building consensus, voting, developing policy, and identifying stakeholders. The GSS software was installed at over 40 universities and colleges and at more than 12 corporations, including IBM, Dupont, BellSouth, and Greyhound Financial Corporation (Jessup & Valacich, 1993).

The software, GroupSystems V™, offers a tool set to support a wide variety of group processes. Some of the tools are discussed below:
1. Agenda: allows groups to pre-plan the flow of the meeting in advance.
2. Electronic Brainstorming: enables the group to gather deas and comments in an unstructured manner.
3. Topic Commenter: supports idea generation in a structured format.
4. Categorizer: enables the group to consolidate and categorize ideas and comments generated.
5. Vote: supports group evaluation of issues through a variety of quantifiable methods.

1.2. The Opportunity

With such a state-of-the-art facility, the MIS faculty knew we could not afford to teach the way we used to; we must assess what we have been doing in the classroom and use the RTR to help us teach better and help our students learn better. It was a great opportunity to find new ways to expand our horizon together.

2. The Round Table Room and MIS

So far, several MIS courses have introduced our students to the RTR whenever the appropriate situation arises. The following discussion will highlight an example of the use of the RTR in each course. Average time per session in the RTR was 50 minutes.

2.1. MIS 16

This 3-credit course introduces students to programming concepts using the Pascal programming language. At the beginning of each programming language course, we always discuss the problem solving process and its relation to the programming process. This spring semester, we were able to bring the discussion into the RTR.

We used the *Topic Commenter* tool to solicit comments on two questions: "What is programming?" and "What do we do in a problem solving process?" The students were able to switch from one question to another to examine what others had submitted, submit their own comments, and/or comment on what others had submitted. Then the comments relevant to the first question were copied into one *Categorizer* tool and the comments pertaining to the second question into another. Using the *Categorizer* tool, the students grouped the comments into main categories, such as 'Define the problem' and 'Generate alternative solutions." Then the main categories were exported to the *Vote* tool where a rank-order method was used to allow the students to organize the major categories into a logical sequence. Each student submitted his/her logical sequence anonymously and the system tabulated all the submissions and presented a "group" sequence. Based on the result presented, the class discussed the sequence shown and whether the class has a strong consensus. The technology enabled the students to be active learners throughout the process. Everybody had an equal say and equal vote.

2.2. MIS 20

This 4-credit course enables us to introduce our first- and second-year students to the MIS discipline. It provides an introductory knowledge of computer literacy, software literacy, basic computer and information systems, problem solving and program development, systems development, data communications and data bases, applications and societal implications of information technology, computer ethics, ergonomics, and career opportunities

In the past, we divided the class into groups of three's or four's when we assigned case studies. It is quite logical to assume that having over 20 students working on the same case and conducting a fruitful discussion is impossible. Some students won't be heard or may not contribute at all. Besides, only one issue can be addressed at a time and the number of comments generated would be low. So, the RTR was used to facilitate our case analysis process.

A case involving the development of a personal information system was posted on the server for all MIS 20 students to read. A week later, the class met in the RTR. Questions pertaining to the problems identified in the case, the development process chosen, the alternative solutions presented, the ethical behavior of the parties involved, and the terms of the contract were posted in the *Topic Commenter* tool. The students could work on the questions in any order he/she wanted. The topic on ethical behavior generated the most discussion. It was a very interesting by-product of the activity; it showed which topic in the case captivated the students the most. Since each topic kept a count of the number of comments submitted, it enabled the students to quantify the level of interest in that topic. Next, the students used the *Categorizer* and *Vote* tool to formulate and organize respectively a series of steps they would use to minimized the problems that occurred during the development process.

2.3. MIS 55

This 4-credit course offers our third-year MIS majors a study of the role of computer information systems, such as information reporting systems and decision support systems, within organizations. One of the topics introduced in this course was stakeholder analysis. We felt this might be a good exercise in the RTR.

The *Brainstorming* tool was used to solicit a list stakeholders of the college. Besides identifying the stakeholders, the students must briefly describe the stakeholder's interest in the college. From the list generated, the *Categorizer* tool was used to compile a list of major stakeholders. Then the list was copied to the *Topic Commenter* tool where the students discussed the interests of the major stakeholders in more detail. After this discussion, the list of major stakeholders was exported to the *Vote* tool where the students then assessed the relative importance of the major stakeholders, using a rank order method. Some of the major stakeholders identified were students, faculty and staff, parents, alumni, administration, local businesses, people in the community, and the church.

2.4. MIS 62

This 3-credit course offers our senior MIS majors an overview of the information systems development process. Besides understanding the role of information system in business, another responsibility of a good MIS professional is to understand the needs of the clients. To do that, good interpersonal skills must be acquired and developed. Interview and research skills are essential in helping the clients define their problems and needs.

Since it will be the 75th anniversary of the department, we saw the opportunity for the students to practice their interview and research skills; their assignment was to chronicle the history of the department. To set up the project, we use the RTR to define the questions they need answers for.

The students used the *Brainstorming* tool to list the potential questions, such as "How did the department get started?", "Who were involved in starting the department?", and "Where was the department housed initially?" Using the *Categorizer* tool, the questions submitted were then grouped into major themes, such as location of department, important dates, professors, and origin of department. Then the class used the *Vote* tool to organize the major themes into a logical sequence to facilitate the fact-finding process.

2.5. Course Evaluation

One of the students suggested that we should use the RTR for course evaluation. What a great idea! The following was an example of how we conducted a course evaluation with the MIS 20 class.

The *Topic Commenter* tool was used to solicit comments on: "What did you learn?" (113 comments in 15 minutes), "What do you like about the course?" and "How can the course be better?" (111 comments in 15 minutes), "How can the professor be a better teacher?" (105 comments in 10 minutes). The *Vote* tool was also used to present 12 items to the students to assess the professor's performance using a 5-point scale. The result was electronically tabulated and a graph was created and shown to the students. The whole session took 50 minutes.

In the past, only the faculty would see all the written comments and suggestions. This time, everybody saw them in real-time. Students were able to agree and disagree with each other's assessment and suggestion. Everyone saw how the class as a whole felt about the course and the faculty.

3. Conclusion And Discussion

The following are typical comments submitted anonymously by students after a RTR session:

"I find it great that we can all discuss multiple topics at the same time."

"This method of communication facilitates conversation for those who may be apprehensive to voice their opinions."

"I can see how it can be useful in making decisions in the business environment."

"I can see where this could aid many different kinds of groups."

"Individuals are more likely to be free in expressing their views. Thus the group would get a more accurate perspective of the situation."

"It prevents some of the group decision-making biases, such as fear of raising opposition and the mistaken belief that everyone agrees."

"It was fun but also a learning process."

"Everyone can get his/her comments and vote in without being self-conscious."

"It should not replace verbal interaction but it is a good starting point for that communication."

"I had a great time and can't wait to do it again."

Our MIS program tries to learn from MIS colleagues in other institutions regarding MIS curriculum and contents (Novitzki, 1990; Stolen, 1992), teaching techniques (Moncada, 1993; Thomson, 1994), development and enhancement of MIS courses (Goldman, 1994; Stemler & Chrisman, 1993), planning for change (Behling & Wood, 1993), the challenges they face (Kim, 1993), and successes they have achieved (Moncada, 1993).

The RTR was a challenge to provide a better learning experience to our students and to engage them in active learning. To do that, faculty must examine the way we teach. "Business as usual" is not going to be enough. We must use the tools presented to us to be better teachers, to help our students be better learners.

It is very easy and tempting for us to embrace the possibilities the RTR technology has to offer without seeing the value it brings to the courses. Technology itself, however, is not enough. There is a greater purpose to learning than what technology can provide. The curiosity which spark a student to investigate a topic, the ability to solve a problem through a logical process, the thirst for new knowledge, and the ability to focus on the "bigger picture" rather than just the issue at hand are all things all faculty should help our students to develop.

References

1. Behling, Robert, Wood, Wallace, (1993), *Successful Planning for a Changing MIS Education*, Journal of Education for Business, vol. 68, July/August, pp. 353-357.
2. Goldman, James E., (1994), *Design and Implementation of a Practical Business Oriented Undergraduate Data Communications Curriculum*, Journal of Information Systems Education, vol. 6, no. 2, Summer, pp. 80-85.
3. Jessup, Leonard M., Valacich, Joseph S., (1993), *Group Support Systems: New Perspectives*, Macmillan, New York.
4. Kim, Jae W., (1993), *MIS Program in Smaller Colleges: Politics and Other Problems, Education*, vol. 114, no. 2, Winter, pp. 271-280.
5. Moncada, Susan M., (1993), *Applying Cooperative Learning to MIS Education*, Interface, vol. 15, Issue 1, Spring, pp. 11-15.
6. Novitzki, James E., (1990), *Scheduling and Content of MIS Courses at the Undergraduate Level*, Interface, vol. 12, no. 2, Summer, pp. 41-42.
7. Nunamaker, Jay F., Dennis, Alan R., Valacich, Joseph S., Vogel, Douglas R., George, Joey F., (1991), *Electronic Meeting Systems to Support Group Work*, Communications of the ACM, vol. 34, no. 7, pp. 40-61.
8. Royksund, Conrad, (1995), *The Round Table Room at Luther College*, Luther College, Decorah, IA.
9. Sandman, Thomas E., (1993-94), *A Framework For Changing a MS/MIS Curriculum to a Changing Environment*, Journal of Computer Information Systems, vol. XXXIV, no. 2, Winter, pp. 69-73.
10. Stemler, Luann, and Chrisman, Carol, (1993), *Introducing Teamwork Through Programming Courses*, Proceedings of the Tenth Information Systems Education Conference.
11. Stolen, Justin, (1992-93), *The Undergraduate MIS Curriculum: A Sampling of AACSB Schools*, Journal of Computer Information Systems, vol. 33, Winter, pp. 54-57.
12. Thomson, Nancy S., (1994), *Using TQM Principles To Teach Current Topics in Information Systems*, Journal of Information Systems Education, vol. 6, no. 2, Summer, pp. 65-69.

Program Advisory Committees: Are We Effectively Utilizing This College Resource

Stephen Mansfield
Associate Dean
College of DuPage
Glen Ellyn, Illinois

Occupational/Vocational/Program Advisory Committees are mandated at the local level in order for community colleges to receive state and federal funding for these programs. Are schools effectively utilizing the advisory committee as a resource for program development, improvement, and assistance?

The role of the advisory committee has been described as providing a linkage between the educational agency and industry and as a communication channel between the college and the community. This collaboration is successful as long as the role of the committee is advisory.

With the change in our economy to a global marketplace, it is increasingly important for schools to be attuned to the needs of local, regional, state and national employment needs. Publications such as "A Nation at Risk" and "High Skills or Low Wages" emphasize the importance of an educational system that is capable of providing a workforce prepared for the challenges of the twenty-first century.

The accountability and continuous program improvement issues that face many of our campuses should be a major impetuous for us to look to program advisory committees for providing insights and assistance in ensuring quality occupational and vocational curricula. The program advisory committee can provide us with vital information regarding current and future employment needs as well as specific skills necessary for entry into a given career path. By meeting the legitimate needs of our community, we likewise fulfill our commitment to our students by offering relevant and up-to-date occupational programs.

The use of the ceremonial and/or rubber stamp program advisory committee is a disservice to its members and the college.

Studies of advisory committees have suggested several common activities which they should perform. Activities may include: assistance in development of good community public relations; advice in development of new or revised occupational course content; utilizing community resources to support instructional programs; reviewing occupational programs and providing suggestions for program improvement; assisting staff in conducting occupational and/or community surveys; making recommendations regarding requirement and facility planning; assist staff with student placement; assist administration with locating qualified faculty.

This presentation will include a brief overview of some of the research that has been conducted on advisory committees along with a bibliography of relevant publications. Information regarding suggestions on "how to" effectively organize and utilize the program advisory will be shared with participants.

Audience participation and input is encouraged.

Globalization Issues in Information Systems Education: Toward a Collaborative Multischool Systems Analysis Experience

William J. Tastle
Dept of Mathematics and Computer Sciences
Ithaca College
Ithaca, New York 14850

U. Rex Dumdum
Dept of Business and Managerial Science
Marywood College
Scranton, Pennsylvania 18509

Abstract

The problems and complexity associated with globalization directly impact the Information Systems curriculum, especially with respect to the formation and management of teams of systems analysts. Though it is not feasible, nor desirable, to provide instruction to IS students in how to relate to all cultures when confronted with team membership whose occupants possess differing skills, cultures, and beliefs, a suitable experience can be given to students in the Systems Analysis class. Extending beyond the usual set of well-defined, unambiguous in-class problems is the external "real-world" problem in which complexity and ambiguity reign in problems stretching beyond traditional borders and into the global marketplace. To provide the Systems Analysis and Design class with a simulated experience of working in the global environment we have utilized actual problems from the commercial, governmental, manufacturing, and nonprofit industries. To experience these situations and provide for the development of some expertise in dealing with these problems, students are placed into teams and given the responsibility for problem solving to the satisfaction of the industry principals. Two types of student teams are identified: homogeneous or single school teams, and heterogeneous (dyad or triad) school teams. Homogeneous teams share common instruction, a common body of knowledge, and inter-team commitment and accountability, while heterogeneous teams find incompatibilities in their basic level of shared and unshared knowledge, CASE tools, methodological approaches to problem solving, commitment to solving the problem, and team accountability. Homogeneous team experiences are useful in establishing team work habits and allowing students the opportunity of dealing with known personalities, and heterogeneous teams extend that experience to include opportunities involving unknown individual personalities, intra-team commitment and accountability, and the pressure of deriving an acceptable solution regardless of obstacles. We suggest this experience can be used to satisfy portions of sections 2.4, 2.10, and 3.7 in the IS'95 Model Curriculum.

1. Introduction

Globalization is one of the most powerful and pervasive influences on nations, businesses, workplaces, communities and lives of the late 20th century. Kanter [1995] claims that organizational success in the economy of the 21st century will only come to those organizations whose goods and services meet world class standards and which can compete in the global marketplace. Others concur [Ives, et.al. 1996; Mitroff 1987; Hamer and Champy 1993] and suggest that American business must increasingly and effectively compete in a global environment in which extreme interconnectedness, flexibility, innovation, quick response, and focus on process rather than task, is a fundamental feature.

Is this just more hype, an opportunity to publish more books and papers, or should IS educators pay heed to this latest impetus? To be sure, the problems and issues of globalization in the organization are not new, for John Naisbitt [1982] presented them more than a decade ago. He recognized the change from a national economy to a world economy, from hierarchical performance to performance by network. Katzenbach and Smith [1993] concur with this assessment and indicate a trend that most work in modern organizations will be accomplished by high performance teams. The use of teams to solve complex problems clearly outperformed individuals acting alone or in larger organizational groupings, especially when performance required multiple skills, judgements, and experiences [Katzenbach and Smith 1993]. Further, team-based collaborative processes allow parties who see different aspects of a problem to constructively explore their differences and search for solutions that go beyond their own limited vision of what is possible, to resolve conflicts, to enhance commitment, and to construct and advance shared understandings, meanings and visions [Gray 1989; Gause and Weinberg 1989]. Team performance, however, may be hampered across cultures. For example, in a recent study

[Robey and Rodriguez-Diaz 1989] reporting on an implementation of a system across cultures in Panama and Chile, the critical elements which led to project failure were cultural incompatibility and the lack of shared meanings among the members of the responsible teams. Certainly, the level of complexity increases when the context of domain is global. In response to the question we raised above, the response is a definite "Yes, we ought to pay heed to the influence of globalization on the IS curriculum."

The training of IS graduates for employment in multinational corporations has seemingly not been sufficiently addressed by the IS'95 proposed curriculum for information systems programs, although there exist several elements in the body of knowledge which lend themselves to the placement of global concerns, specifically sections 2.4 Organizational Behavior, 2.10 Interpersonal Skills, and 3.7 Project Management. Certainly, employment in multinational companies implies that problems which very likely will be directed towards our IS graduates are large, very complex, and evolutionary in the sense that the original problems or views of problems will expectedly change during that portion of time during which the requirements analysis is conducted. The latter point is one which we (as systems educators) address in the typical software engineering and systems design courses, that of designing a solution and then implementing it in a human activity system. The solution, though it might appear as ideal in a classroom environment, hardly seems to satisfy the needs of such a project if it was derived from an ongoing business; this lack of success in the business environment is addressed in the form of system upgrades. The upgrade of a real-time, dynamic human activity system requires, by definition, that the solution be, at best, on the trailing edge of current needs. Such a characteristic may well be endemic to systems design.

2. Issues for Information Systems Educators

As business, industry and government refocus and reinvent themselves to meet the trends towards globalization and team-based high performance organizations and its corresponding requirements and challenges for the 21st century, *how must the IS academic environment evolve to be proactively dynamic* such that its curriculum is equally relevant? Also, *what types of learning experiences* must we fashion to effectively meet the demand for globalized team based high-performing IS professionals? To be sure, the IS'95 Model Curriculum is a giant step in the right direction; its design as a minimum guideline by its authors allows for the fine tuning of an IS curriculum to serve the unique needs and resources of each school.

The problem of attempting to solve very large systems in an active, real-time environment, is, however, a continuing dilemma which we feel has not received a sufficient quantity of study. *The academic projects assigned in a typical systems analysis class are well defined, artificially constrained, and fully known* [Tastle and Dumdum 1994]. Thus, any project given in the classroom environment is at best a highly controlled simulation of the complexity involved in the real world.

This requires us to address the issue of how we can provide a collaborative learning environment for information systems students when they share a single classroom and the problem(s) are academically determined. The authors have successfully addressed this problem over the past six years within the context of the Systems Analysis and Design course and are expanding the classroom situation such that a multinational "experience" can be derived within the limited confines of most academic institutions. Furthermore, the IS'95 Guidelines clearly argue (page 5) that

> "Graduates need to be able to interact more effectively with clients and to work more effectively in teams. They need more depth of technical knowledge in dealing with a wider set of technology. These changes dictate an increased level of resources in terms of hardware and software, more knowledgeable faculty and improved pedagogy."

Implied within this charge is the presence of a set of qualities or competencies which should be present in all IS graduates, particularly in view of globalization.

The issue of competencies in IS education has been previously established [Tastle and Dumdum 1994;Dumdum and Tastle 1996] as being apportioned into three overlapping components: technical, conceptual and interdependency competencies. Technical competence is defined on two levels: (1) "tools" competency which involves a knowledge-base of various hardware and software products at a reasonably sophisticated level of detail; and (2) "transformative" competency which involves the ability to apply information technology as a transformative force for shaping, supporting, achieving, and maintaining strategic corporate initiatives [Hamer 1990; Porter 1996] such as the rapid development and deployment of quality products and services [Morgan 1988], the reengineering of business processes [Hamer and Champy 1993], and the overall performance improvement of the organization [Sprague 1993, in Longenecker 1995].

Conceptual competencies consist of specific mind sets and conceptual skills. According to Kanter [1995, 1983], some mind sets are more restrictive than others. For example, she contrasts an "integrative" mind set to a "segmentalist" mindset to approaching problems. To be integrative, a vital key to innovation, is to be willing "to move beyond received wisdom, to combine ideas from unconnected sources, to embrace change as an opportunity to test limits...to see problems as wholes, related to larger wholes, and thus challenging established practices rather than walling off a piece of experience and preventing it from being touched or affected by any new experiences." In contrast, to be segmentalist is to compartmentalize actions, events, and problems in keeping each piece isolated from others, thus seeing problems as narrowly as possible, independently of their context, and independently of their connections to any other problems [Kanter 1983]. In a later work, she further contrasts two other mind sets: a cosmopolitan or global mind set and a local mind set [1995]. She asserts that cosmopolitans are "comfortable in many places and able to understand and bridge the differences among them, possess portable skills, and a broad outlook." In contrast, while some people are widely traveled, their mind sets remain parochial, such people being referred to as possessing "local" mind sets [1995]. Conceptual skills include, among others, reframing problems to create innovative solutions, translating vision into action steps, dealing with multiple stakeholders representing conflicting interests, and managing change, complexity and ambiguity.

Interdependency competency, within and across corporate and ethnic cultures as required by a global view, consists of two parts: (1) the ability to build bridges and alliances among various stakeholders [Morgan 1988] and (2) the ability to actually collaborate in high performance teams to deliver world-class results [Katzenbach and Smith 1993]. The first accents the need to be able to establish and maintain personal and/or professional "connections" - the best relationships - so that one may have access to the resources of other people and organizations [Kanter 1995]. The second signals the ability to collaborate with team members to explore, evaluate, and arrive at shared meanings and understandings, identify a common purpose, establish performance goals and approaches, and to productively translate mission into high quality results. This competency is not as strongly emphasized in past and present IS curricula for it is experientially based, and industrial/commercial/governmental/non-profit professorial experience is usually the exception in the average IS department. The IS'95 guidelines state that IS graduates should have skills in three levels (see page 5 and 7 of the IS'95 Guidelines), and these map very nicely into our three competencies.

We have begun a process whereby our students have developed an appreciation for, and greater knowledge of, the challenges involved in the problems of team-based high-performance demands emanating from the Systems Analysis and Design course [Longenecker 1995, section 95.7].

3. Classroom versus Industry Experience

The available cases for classroom experiences are replete in the plethora of systems analysis textbooks for each textbook has its own collection of case studies, projects, learning experiences, etc. Each case is usually complete and unambiguous, as one would expect from any well designed project, but this is exactly the type of problem that the graduates will not experience, though the knowledge learned and the experiences derived from these academic problems are, of course, better than no project experience at all.

Over the past twelve years we have established a network of contacts in the industrial, commercial, nonprofit, and governmental sectors of our surrounding counties and can now relatively easily find problems composed of a suitable degree of complexity and ambiguity such that we bring into our classes an "actual" experience.

Typically we partition the semester into two blocks. The first block comprises the first ⅓ to ½ of the course and is devoted entirely to a very compressed and demanding presentation of traditional concepts of analysis and design. Most of the material contained in texts are covered in very rigorous format. It should be mentioned that an equally rigorous course in database theory and application is a prerequisite for the systems analysis course. This nontraditional sequencing of courses has provided the students with the knowledge necessary to more adequately focus on the design methods and not to be sidetracked with excursions into database theory. We have found this sequencing to be very beneficial. Further, small abbreviated problems, such as traditional case studies, are assigned as team projects. More recently we have found that assigning students a known functional area (i.e., the borrowing of a book from the college library) to analyze has yielded a successful experience. Students are usually reluctant to ask questions when first confronting a corporate president or manager and consequently use the instructor as the questioning initiator. In time, the students take control over the investigation, but they do feel unsure of themselves. We have sought to reduce this feeling of insecurity and have succeeded in large part through the analysis of on-campus departments with which our students already have established a comfort level.

3.1 Academic Projects

Typically, our first project might be the analysis of the advance registration process, one in which each student already has an intimate knowledge, or so they believe when first presented with the problem. A representative of the Registrar's Office leads a very detailed discussion of the process about which the students quickly discover they actually know very little. The resulting entity-relationship, functional, and data flow diagrams are created by teams, corrected by the instructor, and discussed in class. It is also during this time that the students are introduced to a CASE tool. Our experiences with CASE tools is such that neither of the authors wish to discuss the shortcomings or merits of the several tools we have experienced in the past, but it is perhaps noteworthy to comment that the recent packaging of student version CASE tools with the systems texts seems to be a step in the right direction, although that step is arguably a small one. The ability of students to use their own computers to create diagrams and the concomitant data dictionary is very useful.

Another project which has met with varying degrees of success is the analysis of the campus security department. Other than the registration of cars most of our students have had little contact, so a meeting with the director of one of our college's security department (a former FBI agent) invoked a considerable measure of stress on the students, though they are better able to enter into discussions and have developed some degree of confidence in themselves and their knowledge. These projects are analyzed from the traditional structured analysis point of view and/or the object-oriented perspective.

3.2 Real-World Projects

Additional projects have involved the college library operations and business aspects of the college, and through each experience, the students' ability to apply systems concepts is enhanced. Around mid-semester the "real" project is presented and discussed. This initiates the second block and focuses the remaining course on the analysis and design of a very complex and ambiguous project located outside of campus, though classroom lectures and laboratory exercises continue but at an abbreviated rate. The location of a project off-campus is not to imply that suitable projects cannot be found within the collegiate environment, for some of our colleagues are able to supplement their systems courses with projects from within the college, but external to the academic arena. The main project is usually located about an hours' travel time from the college so that the students cannot arbitrarily travel to their client at whim but rather, must carefully plan out their time and be prepared to take maximum advantage of their opportunity to meet with client officials. This limitation is artificially imposed on the students, but we know it is important for IS graduates to be able to express themselves succinctly and completely so we have deliberately legislated this constraint.

Past projects have included: the analysis of a golf course operation; a Fire Marshall's permits and registration system; a local area network operation, usage, and redesign for a nonprofit organization; a patient registration system for a regional health care facility; a Treasurer's Office for a governmental subdivision; a police department; a court system; and another fire department. Waiting for the next class is a problem involving a public electric utility.

3.3 Deficiencies

These projects have been addressed only by students working in teams from the same college, so their experience has been entirely limited to mutually known CASE tools, solution concepts, and an overall shared knowledge base. Students working on these projects have the safety of knowing that all members of their team have the same knowledge and will be seeking to analyze and design solutions in a very similar manner, and that is the main problem with this approach: students are not introduced to problems and opportunities which emanate from being able to work with others who possess different views, experiences, and knowledge. One could perhaps argue that the presentation of a suitable mixture of readings and exercises could allow the students to develop a similar level of expertise. We agree that a similar theoretical knowledge could be derived in this manner, but the opportunity of interaction with another student group is the dominant reason (though not the only one) for this approach.

4. Multi-School Collaborative Academically Simulated Environment (MCASE)

To provide a means by which an MCASE experience might be provided to our students, we have identified a four process method: faculty collaboration, single school student teams, dyad school student teams, and triad school student teams.

4.1 Faculty Collaboration

The motivation for this work began in 1992 as a panel discussion at Muhlenberg College as part of the Eighth Annual Eastern Small College Computing Conference [Fisher 1992] when it became evident that many panel observers were interested in finding ways of bringing realism into the systems analysis classroom. It was generally agreed that the use of textbook cases lacked virtually all of the ambiguity associated with real projects, and both authors echoed their successes moving outside of the classroom. We continued assigning external projects to our systems classes, carefully monitoring their success and the students reactions to the experiences, and have found that these ambiguous and complex problems have motivated our graduates to seek employment with companies concentrating in the field of analysis and design, and have, almost exclusively, not sought employment as programmers or software developers. A computer scientist could easily argue that this is not a desirable service. We argue, however, that a programmer/**analyst** requires one to be first a competent analyst and second, a proficient programmer. We also note that almost half of our graduates go on to advanced studies in the specialized area of analysis and design, either directly after receiving their Baccalaureate or as part-time students while they work in their profession.

4.2 Single School Student Teams

The second phase, that of single school or homogeneous student teams, has been successfully conducted on both campuses. The problems of transportation, excessive instructor time both in identifying projects and scheduling meetings so that the instructor would be present with every team, is quite demanding. Students who complete these projects complain that the work load is excessive and that the instructor demands are enormous, but at the conclusion of the course when the final reports have been written and the presentations given to the corporate managers, the students reflect over a well earned dinner and recognize the extent of their learning - and the fact that they now have been exposed to a nontrivial professional systems experience - and have proudly survived it. This demanding experience has been an asset to each student, whether seeking entry into graduate school or employment.

Within the single school, students view the real-world problem as a critical part of the course, and hence view it as pivotal to making the traditional Systems Analysis and Design course stronger. They reflect greater pride in their work and strive substantially more to complete the analysis as vigorously, accurately, and professionally as their capabilities allow. Their commitment to excellence and dedication to detail is frequently first discovered here. One-fourth of each class regularly admit that the project motivated them to new heights of excellence, for this is frequently the first time that a goal was more than merely the completion of a class assignment. While students may play the grade game with aplomb, when confronted with strangers who are depending upon them to solve their problems, students do rise to the occasion. They continue to show profound commitment to each other and held one another accountable for their portion of the overall study to a standard that is higher than that which the instructor had established. (We defend our individual standards as being reasonable for a specific class and a particular project, realizing that students ought not to be held to the same standard as experienced professional analysts.) When a goal is established, one to which all accept ownership, the degree of learning that occurs, and the quality of work that ensues, is profound.

4.3 Dyad School Student Teams

The third phase, that of dyad school student teams, is beginning in Spring 1997 with the scheduling of the separate analysis courses to run concurrently on both campuses. A problem of sufficient complexity and ambiguity has been identified residing in a company located about midway between the two campuses (the distance of separation is just over two hours by car over a four lane highway). We are changing neither syllabus nor instructional methodology for we seek to determine how a collection of students from different colleges, using different texts, and taught by different instructors, will approach the solution to a nontrivially ambiguous problem when they are thrust together. The approach can occur in two ways: teams may be composed of students from the individual schools (homogeneous teams) assigned to solving the same aspect of the overall problem, or the teams may be mixed by school (heterogeneous teams). We feel that the latter will, of course, more closely approximate the real-world experience we seek, but we also recognize the difficulty of students traveling to discuss and analyze their assignments. We are considering the use of list servers and teleconferencing to allow for the creation of mixed groups. If we deem this experience to be successful, a nearby third college (a university) will be invited to participate.

In a manner similar to that of the single school experience, we expect our two schools to exhibit an even stronger degree of commitment and accountability towards the completion of the shared project, for we expect that it will be perceived as something of a competition between schools, not unlike the competition between company teams. If it is possible to mix the teams, a stronger sense of commitment and accountability towards the satisfying of the client's problem is expected to be identified.

4.4 Triad School Student Teams

Upon the successful conclusion of the dyad team experiment, we anticipate the addition of another school to the MCASE experience, a major University center supporting an MBA in MIS. The student mix will then become about as close to that of a multinational work environment as is reasonably possible. This will be a very important part of the experiment for the difficulties of coordinating the students of three schools are expected to be exponentially greater, and we currently have no knowledge that the consequent student experience will be worth that effort. This experience should allow us to determine the relative importance of dyad versus triad teams, from which recommendations for future IS curriculum guidelines may be secured.

The dyad and triad school experiences are the subjects of other papers.

5. Conclusions

Multinational corporations seeking to compete in the marketplace have acknowledged the need to address systems problems from a global perspective. Problems are substantially more complex and require the services of teams of analysts, teams whose composition is heterogeneous with respect to education, frameworks, methodologies, knowledge of CASE tools, experience, culture, social and personal behavior, beliefs, and virtually every other attribute, but who must nonetheless work together to solve the current problems. As IS educators, it is our responsibility to provide some guidance and training to our students to help prepare them for this formidable task, but it is unreasonable to expect that we should provide the training necessary to accommodate students being able to work with every type of individual in any global environment. Such an undertaking would be unreasonable if not inappropriate for the classroom. However, we can provide them with some experience and insight into the "global" analysis team environment by providing students the opportunity to address real world problems first as competing teams and second, as partners with students from other schools. The inter-school project strengthens section 2.4.2 Cultural diversity, 2.4.3 Group Dynamics, and 2.4.4 Teamwork, leadership and empowerment, as outlined under section 2.4 Organizational behavior in IS'95, as well as portions of 2.10 Interpersonal Skills, and 3.7 Project Management.

Bibliography

Fisher, Craig [1992], "Systems Analysis and Design: Approaches for the 1990's." Panel Discussion, Proceedings of the Eighth Annual Eastern Small College Computing Conference, Muhlenberg College, 8(2).

Gause, Don and Gerald Weinberg [1989], **Exploring Requirements: Quality Before Design**. Dorset House Publishing, New York.

Gray, Barbara [1989], **Collaborating: Finding Common Grounds for Multiparty Problems**. Jossey-Bass Publishers, San Francisco, CA.

Hammer, Michael and James Champy [1993], **Reengineering the Corporation: A Manifesto for Business Revolution**. Harper Collins Publishers, New York.

Ives, Blake, Sirkka L. Jarvenpaa and Richard O. Mason [1993], "Global Business Drivers: Aligning Information Technology to Global Business Strategy." IBM Systems Journal, 32(1).

Kanter, Rosabeth M. [1983], **The Change Masters: Innovation and Entrepreneurship in the American Corporation**. Simon and Schuster, New York.

Kanter, Elizabeth M. [1995], **World Class: Thriving Locally in the Global Economy**. Simon and Schuster, New York.

Katzenbach, Jon R. and Douglas K. Smith [1993], **The Wisdom of Teams: Creating the High-Performance Organization**. Harvard Business School Press, Boston, MA.

Longecker, Herbert et.al. [1995], **IS'95: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems**. AIS, ACM, DPMA Publishers, USA.

Mitroff, Ian I. [1987], **Business NOT as Usual: Rethinking Our Individual, Corporate, and Industrial Strategies for Global Competition**. Jossey-Bass Publishers, San Francisco, CA.

Morgan, Gareth [1988], **Riding the Waves of Change: Developing Managerial Competencies for a Turbulent World.** Jossey-Bass Publishers, San Francisco, CA.

Naisbitt, John [1982], **Megatrends: Ten New Directions Transforming Our Lives**. Warner Books, New York.

Porter, Michael E. [1996], "Changing Patters of International Competition," in Deans and Jurison (Eds), **Information Technology in a Global Business Environment: Readings and Cases**. Boyd and Fraser Publishing House Company, Boston, MA.

Robey, Daniel and Andres Rodrigues-Diaz [1989], "The Organizational and Cultural Context of Systems Implementation: Case Experiences from Latin America." Information and Management, 17.

Tastle, William and U. Rex Dumdum [1994], "Information Systems Competencies for the 1990's: Issues and Challenges." Proceedings of the Tenth Annual Eastern Small College Computing Conference, St. John Fisher College, 10(2).

Effective Information Systems for International Business

Dr. Longy O. Anyanwu
Associate Professor
Information & Decision Sciences
Montclair State University
Upper Montclair, NJ 09043
E-mail: anyanwul@alpha.montclair.edu

Abstract

The growth of worldwide competition and expansion of business markets especially in Eastern Europe and the third world have triggered several formidable problems for the global information systems(GIS) manager. Although the implementation of client/server systems has been a popular solution alternative to these problems, numerous multinational corporations have failed to meet their objectives beyond their national borders. This study identifies the major recurrent problems and issues, discusses their symptomatic characteristics and impacts on the corporate business and information systems management, streamlines appropriate recommendations for combating each problem, and finally, develops a matrix of issues, characteristics, domain classifications, and suggested recommendations for easy reference for managers.

Introduction

The information explosion coupled with technological advances and the increasing globalization of the world economy, have fostered the continued growth of transnational organizations.[20] Numerous organizations, for strategic advantage, have already become involved in multinational businesses, carrying their businesses across their borders. Even many more are planning to do so in the near future.[11,1] Tremendous opportunities, therefore, have been created for many organizations and formidable hindrances have been amassed against others. Business failure rates among these transnational organizations have increased. What are the impacts of business globalization on the multinational corporation(MNC) in general, and the global information systems(GIS) management in particular? A GIS is a distributed data processing system that is spread in different countries. A few recent studies indicate that GIS managers are fast resorting to client/server(C/S) technology for effective implementation of GIS.[1,8] Looking for new cost effective implementation of multinational IS, to maintain central control at business headquarters while providing for flexibility to adapt to local business culture(s) reminiscent of multinational business activities, the IS managers found in C/S a solution. How can an IS manager successfully implement a C/S-based information in a global environment?

When a company goes global, management begins to face additional problems. It begins to face multilingual and multicultural climates, different legal systems and governmental regulations, different political environments, and widely varied bureaucratic processes. It will find different currencies, multiple time zones and many different approaches to business information systems implementation and education. When a firm enters the global market, all managers struggle with the severe strain on the organization. However, it is the GIS manager or management information systems(MIS) professional, who additionally, is expected to alleviate many of these organizational problems generated by the globalization initiatives. Adding to the difficulties of C/S-based MIS, is the problem of rapidly changing technologies within both the computer and telecommunication industries. Also, the infancy of the C/S technology complicates the situation even further. The would-be-successful managers must not only be able to cope with present problems but also must know when new technology is needed to enhance the operation of a GIS.

Purpose and Method

This study will focus on identifying the issues at stake to managers in the implementation of global C/S-GIS among the MNCs. These issues will be examined in four broad categories or domains with respect to their impacts on business and IS management, namely: general management issues, IS management issues, cultural issues, and environmental issues. Based on the identified problems and issues, the dominant factors, and the results of the issue analysis, this study will streamline a set of recommendations for successful IS implementation and management in an MNC. These recommendations will enable C/S-GIS managers and professionals to adequately deal with these very recurrent problems and issues. It is postulated that through the identification and analysis of these issues, the MNC IS managers will become more aware of them. Additionally, the C/S-GIS implementation and management guide will enable GIS managers, who are exposed to it, become fully equipped to deal with these issues which can sometimes be the sole factors that determine the success or failure of a global business venture. If the C/S-GIS managers are adequately prepared to deal with these issues, C/S-GIS implementation costs and global business failure rates will certainly be reduced

Survey of Literature

For their inherent advantages, C/S systems have, in an unprecedented manner, rapidly swept the IS implementation and operation in businesses in many parts of the world. Several recent studies indicate that more and more businesses have either implemented or are planning to implement C/S system within the next few years. [2, 3, 5, 6] This C/545 growth has, to a large extent, been limited to national business operations. [2] However, as corporations compete in international markets, attention has begun to shift to the utilization of C/S technology to enhance competitive advantage beyond the national borders. [3] Now, a large number of these organizations are embarking on global business initiatives.[1, 7, 8] Also, there is a growing recognition in the literature that managing IS in an international environment poses unique and difficult challenges. [21, 22, 23] Companies want their managers, usually referred to as decision makers, to understand the needs of the global customer and the global environment early in their careers. [7] Numerous studies appear to cluster their data analyses along two concerns: the importance of globalization in organizations and the role of the information technology in its management, and the corresponding reflection of this internationalization in the business school curricula.[13, 14, *15*, 16, 17, 18] Additionally, beside the Data Processing Management Association(DPMA) and the Association of Computing Machinery and Institute of Electronic and Electrical Engineering's(ACM/IEEE) emphatic recognition of diversity (and inclusion of global characters of businesses) in business/IS curricula, the Association for the Accreditation of Colleges and Schools of Business(AACSB) in its April 1992 report on standards for the accreditation of business schools and their curricula, clearly underscored the importance of ethical and global issues in business curricula. While the two biggest barriers to C/S system implementation in 1993 were tools, and skills and retraining, in 1994 the two biggest barriers were systems management and management of change, according to users at the 1994 Uniform Technology Managers' Conference sponsored by Hurwitz Consulting Group, Inc. [4] This shift of problem perception to IS management underscores the increased concern in this area. Some of the recurrent issues in the literature include: communication, decision making, data, and management, technology being the driving and enhancing force in all four barrier areas. [7]

The Major Issues

The distribution of the numerous end-users in a myriad of transnational organizations across the national borders is often uneven with respect to the end-users' socio-cultural heritages, political affiliations, legal and economic environments, and levels of technological know-how. In their attempt to develop, control and directly use the information systems, these users interact among themselves and with this complex variety of systemic differences. This uneven mixture of end-users increases the complexity of the problems traditionally faced in the management of end-user computing, namely, information integrity and security, information privacy and accessibility, and information management efficiency. In addition to these traditional issues, there are a few contemporaneous and increasingly recurrent issues. Based on literature, these emerging issues include cultural differences, user needs and satisfaction, information management efficiency, communication misfire, economic instability, regulatory standards, resource availability, host country demands, network and system security, and system outsourcing. Interestingly, these issues can be either business hindering factors or motivating ones, and some can even be both depending on how they are managed. Following the discussion of each factor, a recommendation to the C/S-GIS manager is made as to the more effective and safer control approach to this problematic business environment. Finally, a matrix of problem/issues, characteristic concerns, major causative factors, and suggested solutions is developed for easy C/S-GIS managerial reference.

Cultural Differences

The degree of cultural homogeneity or heterogeneity in the C/S-GIS professional/user team affects the dynamics of the team and its ability to achieve results. Cultural issues affect the perceived relevance of the task facing the team and how it uses available resources such as time, money, information, technology, etc. People interpret messages and instructions in the context of their cultural heritages. Cultural diversity can be a complex problem as well as critical strength in the survival of a business organization. If, and when cultural diversity is properly managed, problems are avoided or at least minimized, and strengths are exploited to the maximum advantage of the organization. A C/S-GIS manager must have a global perspective. The key is the recognition of diversity. Understanding cultural differences among C/S-GIS professionals is essential.

User Needs and Satisfaction

What are the interests of users, and how can those interests be determined? Unfamiliarity with the users' beliefs and value systems, and their consequential effects on employee motivation and job satisfaction are a major hindrance to productivity and effective C/S-GIS management. How will an organization, in its strategic posturing, lock in to its customers/users who now have widely differing value systems? How does a manger draw up a road map to satisfy all these customers? Will he do well to selectively pick what customers in what countries to satisfy? A broad and participative view of the situation is preferred. A task force of international users, C/S-GIS professionals, legal experts, etc. should be created to search for a balance of user needs and interests versus the business objectives of the firm. Accept the fact that different levels of technological implementation, workable local solutions, and sensitivity to people in each area can bring about a satisfying situation of global competitiveness.

Information Management Effectiveness

The often parochial approach to management of information technology (IT) in many MNCs has become a major liability to the achievement of the organizational strategic objectives. Cognitive and communication styles of management that often make user interfaces more or less acceptable and effective are important issues to grapple with. other information management sub-issues include: managing the acquisition of end-user C/S computing technology for the purpose of transborder integration, compatibility and economies of scale while recognizing local differences in vendor viability and support; and the variety of definitions for "desirable" in decision making and conflict resolution; and the differences in people's attitudes and behaviors which determine leadership styles. Additionally, according to Regina Bento, a particular problem with leadership in cross-cultural teams is the issue of cultural dorminace -the "disproportionate power vested in members of one culture over those of other cultures".[19]

Managing change is, and should continue to be, a focal point in CIS-GIS management. Managerial control through coordination is the solution. Coordination of tasks involve: 1) the analysis of how similar or linked activities are performed in other countries; 2) management of the exchange of information and information technology; and 3) the sharing and use of information on the firm by its different facilities. The target objective in the C/S-GIS management by coordination should always be to enable: i) flexibility in response to competition in different countries, ii) effective scanning of markets around the world, iii) operational effectiveness in the business organization, and iv) preservation of diversity in final products and production location. In recent times, the advances in technology have greatly reduced the coordination costs by reducing the communication and information processing and delivery time and costs. The GIS manager must think global. Any theme must be considered from diversity and global perspectives. The issues are not necessarily mutually exclusive, and neither are their solutions. Figure 1 below shows the natural interaction of the major inherent factors in the management of a GIS.

Communication Misfire

It is said that only about one tenth of everyone's culture (the major reason why each person behaves the way he does) is "visible" on the surface. Communication is a critical managerial skill, and even more so for the GIS manager. It is also critical to the functionality of the CIS-GIS team and the over all business productivity. The lack of much of it has become a problem recently with many MNCs that baffle with marginal operationality with users or employees of mixed cultures. Interpretations of the elements of communication are often superficial without a knowledge of the underlying culture. The ability or inability to communicate in the local language can alone determine the success or failure of a business venture, since communication, whether internal or external to the organizational environment, is not only key to management success but to business success as well. It is impossible to penetrate another culture, to comprehend the differences in values and beliefs, without knowing the culture's language. Otherwise there is only parochial interpretation of communications. The language variations carry with them unique implications on the information exchanged. Computer programs should be written to be language independent. Alternative communication channel(s) may be sought for more effective tele- and nontele-communication.

Economic Instability

Political and economic instability, which often precedes economic insolvency in most of the third world, is a critical business and C/S-GIS management concern. For instance, How does a manager, from a business perspective deal with an internationally isolated and protem dictatorship, which has triggered economic depression on the host country? How does the firm with little or no bargaining leverage handle host country investment nationalization initiatives? Variations in exchange rates can be a major problem and must be tracked. This problem is even complicated further by frequent economic instabilities around the world. Currency-robust systems should be developed.

Regulatory Standards

Working within regulations and standards is not new to the C/S-GIS manager. These regulations and standards are often and conveniently built into the software. However, when the regulations and standards vary with country and proliferate even at the spur of the moment, they become a liability. How do you articulate business or C/S-GIS management policy which will incorporate all the different national regulatory standards pertaining to business activities and information access and distribution without compromising on the organizational business discipline and information security and integrity? Regulations governing types of information and technology or the use and dissemination of them certainly differ in many countries. Users should be expected to make these kinds of local adjustments to enable full compliance to regulations. For instance, there are as many methods of taxation as there are countries around the world. Even in one country, there may be several methods as well as levels of national, regional, and local taxation. Method-adaptable multilevel taxation Systems should be developed and used. Modify or customize applications to local needs and regulatory requirements. Set up standards, rules, and policies governing the characteristics of data and technology (software/hardware) acquired or developed by the firm. Table 1 indicates the major standards governing the information technology applications.

Resource Availability

Unlike the domestic distributed information systems, the GIS covers more than one country; it is exposed to a wider variety of business environments; faces differing levels of resource availability; and much more encompassing technological and regulatory environments such as standards and transborder data flow. Given that some countries are more technologically advanced than others, it is obvious that they likely have greater concentration of IS and/or technological experts than those other countries. In many countries data may not either be reliable or even available. IS architecture, which often is a function of economic buoyancy may be scarcely available. How does a manager cope with the disparaging technological (hardware/software) platforms and compatibility often found across borders? To what extent will a national deficiency in technological know-how be compensated without the introduction of foreign cultural dominance in the teams? How can a C/S-GIS manager maximize efficient use of resources by minimizing wastes, while availing user resource flexibility to meet their various information needs? The lack of balance in, and sensitivity to, country-specific business practices (usually reflecting past IT investments) renders the shareability of product and business operational information impossible. More particularly, the frustration of C/S-GIS managers in finding country-specific applications of IT has emerged as a barrier to a successful IT implementation. A business approach in testing, accepting and adapting new technologies is essential for a competitive edge in C/S-GIS. Reorganization of data processing to conform with country-specific applications may become advisable. You should be prepared to accept less than perfect solutions in some countries that develop a C/S-GIS.

Host Country Demands

Usually, the economic imperatives pushing a firm to achieve large economies of scale by worldwide product standardization coexist with political imperatives pushing it in a different direction. Government, users and customers in each country want the firm to respond more adequately to local needs. The firm wants to focus on its business objectives and make profits. Occasionally, these divergent forces inevitably put a strain on the firm, and often become unbearable to the firm. Be sensitive to local needs and give them a high priority of consideration. Powerful databases and access languages that are flexible for local and rapid application development should be strongly encouraged.

Network and System Security

One of the traditional functions of the IS manager is to protect the information system, make information available to authorized users, and maintain high information integrity. This goal has not been an easy one to C/S-GIS managers generally. Complicating this problem further is the introduction of the multinational factors into the equation. Battling with threats to information integrity, security and privacy while assuring user accessibility to information, and a sense of local autonomy has been a big headache to MNC C/S-GIS managers. Incrementally leveled technology transfer (where necessary) is the way to go. Software piracy and intellectual copyrights, hardware compatibility, and congruency of database formats must be carefully monitored. Utilize batch transfers of files, reports, orders, etc. between major cites. Incorporate interactive consultation with central database or applications in responses to requests. Use messaging systems that include e-mail and electronic data interchange(EDI) between employees, sites and business partners. The basic rules include: 1) reach out to every user; 2) establish a people network (friendly, culturally sensitive and adaptable); 3) install localizably regulation-adherent Systems (comformable to local rules and policies); 4) persevere to succeed even in the face of adversities or minor failures; 5) maintain a tight security within and without the C/S-GIS system; and 6) endeavor to know all the costs and options before a selection. For extended information accessibility as well as security, remote computing services, data reduction, and data and Systems duplication techniques should be incorporated. Update data and technology frequently. Develop a global data dictionary for all users to follow. A solution to these problems may involve a balance of the strengths of both the centralization and decentralization of the system, rather than one or the other. [19] Table 1 below illustrates the popular C/S~IS control strategies.

Business Strategy/Structure	Coordination/Control Strategy	Coordination/Control Mechanisms	C/S-GIS Structured Strategy
• multinational/decentralized federation	• socialization	• *hierarchies*; material & services flow determined by managerial decisions	• decentralized/standalone C/S-GIS database & processes
• global/centralized federation	• centralization	• *hierarchy*; decision made & control exacted by same managerial unit	• centralization/centralized C/S-GIS databases & processes
• international & inter-org. coordinated federation	• formalization	• *markets*; material & services flow determined by market forces	• linked C/S-GIS databases & processes
• transnational integrated network	• coopting	• *network of units*; representative participation in decision making	• integrated architecture/shared C/S-GIS databases & processes

Table 1: Global Information Systems Control Strategies

System Outsourcing

Outsourcing is another critical issue in transnational business and global information system management. Mehdi Khosrowpour correctly noted that outsourcing has become such an important issue for managers of information systems and technology for reasons which include the facts that increasing number of businesses are facing tougher competition in national and international markets, market share is dwindling while global pressures are increasing, and product life cycles are getting shorter, therefore prompting many organizations' sensitivity to efficiency and bottom-line results. [24] Additionally, there is tremendous shortage of skilled IS professionals, and this shortage is projected to increase even more sharply in the foreseeable future. As much as it is possible, in-house software should be developed. If outsourcing is the approach to systems development, then C/S-GIS management strategy with corporate information systems architecture (ISA) is very valuable in providing a guide for systems development. It also facilitates the integration of, and data sharing among, applications. Another benefit is that it supports the development of enterprisewide integrated data resource Systems. In this area, the responsibilities for the C/S-GIS manager will include: 1) awareness of the firm's business challenges and sharing of the leverage of the IT for them; 2) articulating C/S-based global information Systems development environment that reflects the firm's multinational posture; 3) preparing applications development portfolio that aligns with the firm's global objectives; 4) reflecting the firm's strategic global aspirations in the systems development project goals; 5) acquisition of multiculturally adaptable IT; 6) leading in the automation of the firm's internal and external data communication linkages; 7) designing C/S-GIS databases derived from the firm's value-chain activities; and 8) facilitating corporate restructuring through the provision of flexible business services. Each software should be developed to enable easy fine tuning to local needs while maintaining the same data processing and file format consistency throughout the enterprise.

Conclusion

The identification of the major issues, the analysis and discussion of them, as well as symptomatic description of each of them, no doubt, enhances the preparation of the GIS or C/S-GIS managers for the formidable responsibilities that await them in their transnational business activities. Because all the problems may not necessarily manifest in any one MNC or business venture, the discussions and recommendations are individualized to each issue. As has been earlier discussed, more often than not, the problems or their causative factors overlap or compound one another. Very rarely do they exist in isolation of the other. Thus, although the individualized solutions are adequate remedies for each problem, holistic thinking is the approach. In Table 4 the problems are further classified into cultural, environmental, information systems, general management, and communication issues to enable managers make appropriate selection of solution type. By the elucidation on, and analysis of, the issues, and with the suggested CIS-GIS solution alternatives, GIS managers will become more aware of the problems and how to handle them. The success rates of CIS-GIS implementation and management will certainly be improved and costs reduced

Issue	Domain[1]	Characteristic concern(s)	Suggestion(s)
Work Habits	C	• Team's heterogeneous social & work behavior	• Think global • Plan for cultural diversity • Take advantage of the strengths of cultural diversity
User Satisfaction	M, C	• Meaning of user satisfaction • Differing user value/belief systems	• Create task force to balance user interests & business objectives • Use workable local solutions
Management Effectiveness	M	• Parochial management (mgt)of IT • Mgt. Of tech. Transfer & integration • Cross-cultural dominance in mgt.. teams	• Adopt mgt. through coordination style • Consider themes from a global perspective
Communication Misfire	C	• Incomplete or misinterpreted communication	• Learn of other cultures & languages • Develop language-independent programs • Seek alternate communication channels
Economic Instability	E	• Economic volatility & insolvency	• Currency exchange rates must be tracked • Currency-robust systems should be developed
Regulatory Standards	EM	• Variant proliferation of standards & regulations in host countries	• Set up standards/rules for the acquisition/development of systems by the firm • Method-adaptable multilevel taxation information systems should be developed • Customize applics. To local needs & regulation requirements
Resource Availability	I, E	• Country-wise differential availability of resources (human, data, technology, etc.)	• Use country-specific or country-adaptable applications • Be prepared to accept less than perfect products/solutions in some countries
Host Country Demands	E	• Adequacy of response to local needs	• Consider local needs with high sensitivity & priority • Database languages should be locally adaptable
Network & System Security	I, M E	• Information accessibility • Threats to information integrity & system security	• Use user-friendly systems • Monitor violations to security regulations • Develop global data dictionary • Batch transfer of files, etc. should be used often • Use messaging systems between sites • Update data & technology frequently
System Outsourcing	I, M, E	• Outsourcing in systems acquisition/development	• Where feasible use in-house system development • If outsourcing is used, incorporate the organizational information system architecture (ISA) • Develop enterprisewide integrated data resource systems • Acquire/dev. Multiculturally sensitive or adaptable systems

Table 4: Matrix of Issues and Solutions

[1] "Domain" indicates the major area of user-work-life in which the dominant causative factors of the issue or problem exist. C -for cultural factors, E -for environmental factors other than cultural, I -for information systems factors, and M -for basic management factors. In some instances, there is a domain overlap. In such cases, the domain factors are listed in order of dominance.

References

1 Flynn, Anthony J., "Global Information Systems: Problems, Solutions, and How to Manage Them", Journal of Information Systems Education, 6 3, Fall 1994.

2. Anyanwu, Longy O., "Client/Server Distributed Network Systems Downsizing with Hidden Risk Minimization", University System of Georgia Annual Computing Conference Proceedings, 1994.

3. Schultheis, Robert, "The Impact of Client/Server Technology on the MIS Curriculum", National Information Systems Education Conference(ISECON'94) Proceedings, DPMA Education Foundation, October 28-30, 1994.

4. Alper, Alan, ed. "Client/Server's Top 10", Computerworld Client/Server Journal, May 1994.

5. Kim, Chung S. et al, "A Managerial Perspective on the Information Technology Needs Of EndUsers", Journal of Information Systems Education, 7 1, Spring 1995.

6. Forcht, Karen A. and Joan K. Pierson, "Downsizing of Information Systems... Hardware, Software and Budget Considerations", National Information Systems Education Conference(ISECON'94) Proceedings, DPMA Education Foundation, October 28-30, 1994.

7. Aggarwal, A.K. and Bruce Rollier, "Globalization and the Information systems Curriculum", National Information Systems Education Conference(ISECON'94) Proceedings, DPMA Education Foundation, October 28-30, 1994.

8. Kizior, Ronald J., "Strategic Factors in International Information Systems: The Global Perspective", National Information Systems Education Conference(ISECON'93) Proceedings, DPMA Education Foundation, November 5-7, 1993.

9 Schleich, John F., "Information Systems Study in Hong Kong: An American Ex-patriate's View", National Information Systems Education Conference(ISECON'94) Proceedings, DPMA Education Foundation, October 28-30, 1994.

10. Hoplin, Herman P., "The Huge Leap in the Use of Emerging IS-Related Technologies", National Information Systems Education Conference(ISECON'94) Proceedings, DPMA Education Foundation, October 28-30, 1994.

11. IMR Interview: Peter G.W. Keen, Information management Review, 1989, vol.4 no.3, *pp.65-75.*

12. Deans, P. Candace, et al, "Identification of Key International Issues in US-Based Multinational Corporations", Journal of MIS, 1991.

13. Hedlund, G., and D. Rolander, "Action in Hetarchies-New Approaches to Mapping the MNC". In Bartlett and Hedlund, ed., Managing the Global Firm, 1990.

14. Omahe, Kenichi, "Managing the Borderless World", Harvard Business Reviews, May-June 1989.

15. Reich, Robert B., "Who Is US?", Harvard Business Review, January-February 1990.

16. Senn, James A. "Globalization Means 'Of Unknown Origin'", SIM Network, September-October 1991.

17. Stair, Ralph M., Principle of Information Systems, Boyd-Fraser, 1992.

18. Thurow, Lester, "Who Owns the twenty-first Century", Sloan Management Review, Spring 1992.

19. Bento, Regina F., "Cross-cultural Teamwork in End User Computing: A Theoretical Model", Journal of End-User Computing, 7 4, Fall 1995.

20. Barlett, C. and S. Goshal, "Managing Across Borders", Harvard University Business School Press, 1989.

21. Cash Jr., J.I, et al, Corporate Information Systems Management, 3rd ed., Homewood: Irwin, 1992.

22. Deans, P.C. and M.J. Kane, International Dimensions of Information Systems and Technology, Boston: PWS-Kent, 1992.

23. Lucas Jr., H.C., Information Systems Concepts for Management, 5th ed., NY: McGraw-Hill, 1994.

24. Khosrowpour, Mehdi, Managing Information Technology Investments With Outsourcing, Harrisburg: Idea Group, 1995.

Model-Driven Computer Information Systems Curriculum

W. Don Gottwald, CCP, Walsh College
Anthony D. Giordano, CCP, Coopers & Lybrand

Abstract

The paradigm of Information Technology (IT) in the 1990's is significantly different from the Data Processing model of the 1970's. In the model of the 1970's, systems analysts defined the systems requirements, programmers developed the code, and systems operations maintained the environment. Each area was responsible for a distinct set of tasks and deliverables with minimal interconnect between each other. There was a fair measure of truth in the wry comment of the time that the "systems analyst writes the spec.'s, then throws them over a wall to a programmer." We have modeled our curriculums in the same manner. A student will take a COBOL class and develop a payroll program for a School, then take a Systems Analysis class and learn about accounts receivable for an automotive company.

The IT paradigm of the 90's brings us new roles, and interaction between those roles. Instead of Programmers, Systems Analysts, and Computer Operators, we now have Developers, Facilitators, Systems Architects, Database Administrators, Network Specialists, and Integrators. The roles of the 90's are much more proactive and interdependent than before. We now must rely on rigorous methodologies and project planning tools to succeed in developing the systems the users need with the best-of-breed technologies available. We have been somewhat successful in introducing client/server based technologies in the class room, however the real value-add will come when we can infuse the "big picture" into the students. For that, we look to how this has been accomplished in the industry.

The "Rolling up" Concept

It has been observed in the past that programmers seem to become good systems analysts. At first glance this would seem incongruous. Programmers view the world from a micro-bits/bytes approach, where systems analysts view the world from a macro-big picture approach. However, time and again programmers ***within an*** organization make the successful transition from programmer to systems analyst.

The unrealized differentiator for that individual is that through their programming efforts, they have acquired an understanding of the organization's *business processes.* Newly promoted systems analysts leverage their knowns (understanding of the business processes) against their unknowns (the new skill sets of a systems analyst). This leveraging of business processes is the "big picture" aspect we need to educate to our students.

Business Models

Business processes are a subset of an overall business model. The nature of businesses and particular industries equate to the fact that there are common overall aspects to certain business models.

Example 1

One of the largest, best documented industries is the utilities industry. Every major utility company contains a Customer Information System and a Work Management System. These are common **business processes** in the **same business model**.

Example 2

All the major automotive companies have moved toward a Just-In-Time (JIT) Inventory system and are moving to EDI as a method for purchasing and inventory replenishment. The business processes of JIT Inventory and EDI purchasing will remain fundamentally the *same*, regardless if the supplier produces hubcaps or ball bearings. It also is irrelevant as to whether the company is Chrysler, Ford, or Toyota.

The concept of common business models is core to the recent growth in commercially developed Accounting - Financial packages such as SAP, PeopleSoft, and Oracle Financials. It has also been used by the Big Six consultancies in marketing industry templates. It is time for education to leverage a common business model to assist in the knowledge transfer of technology to our students. By using the same business model in each class in a curriculum we will advance the following important concepts:

1. Faster Technical Knowledge Transfer

By using the same business model in each class, students will begin to be able to leverage their knowns (the business model), against their unknowns (the new technology or method). This facilitation will enhance the overall learning experience for both the student and the faculty.

2. Introducing "Real World" into the Class Room

By looking at the same problem, or aspects of that problem from a different perspective each time students will learn how to best use a technology or method to resolve a business problem. By learning when, how, and what technology to use in resolving the same business problem, students will learn that there is more than one route to problem resolution.

3. Integrating a Structured Methodology into the Curriculum Using a Real World Approach

The use of a common business model will assist in taking students through each of the phases, stages, activities, and tasks associated with real life projects using a structured methodology such as Information Engineering. By having deliverables from one class used as input to another class, students will experience the real world concepts of transition and project interdependencies.

4. Class Objectives

The use of a common business model will provide a "template" of the objectives based on the methodology deliverables, that should be required for each class. This will provide instructors with the opportunity and mobility to instruct other classes in the curriculum. In addition, it will assist instructors with providing a consistent message for each class.

Proposal

To use a common business model in our curriculums that will provide specific problems and required deliverables, for each class that will highlight the usage of a particular technology or method. It is recommended that the curriculum model the tasks and deliverable after a structured methodology, such as Information Engineering or Rumbaugh/OMT, to transfer the knowledge of the business model and the benefits of proper project planning.

Approach

Each class will be furnished with two parts of the model: an introduction/re-examination overview of the model and the specific Method/Technical task that is associated with that particular class.

For an example of how a specific methodology deliverable will interface between classes are:
Process Models developed in a Information Systems Planning and Analysis class can be used as an input for a Programming Lab class.

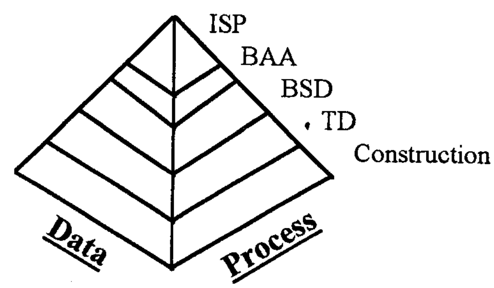
Note: In traditional Information Engineering development the process follows top down. However since the nature of a college program is to instruct students in classes such as programming and then systems analysis, these classes will actually become the building blocks of the structured methodology. This will provide students the ability to leverage their growing "big picture" knowledge of the business model in the later strategic IT classes.

Enclosed is an Introduction to Information Engineering, sample Curriculum to Information Engineering Matrix, sample Curriculum with deliverables, a sample Business Model, and delivery mechanisms required of the faculty in this approach.

Introduction to Information Engineering

What is Information Engineering, and how does it approach systems development? Information Engineering (**I.E.**) is a data driven, structured methodology used for developing and managing Information Technology (**IT**) development projects. I.E. is based on the work of Dr. James Martin in the mid 1980's. I.E. provides logical step by step systems development from an enterprise approach. I.E. approaches systems development as a series of stages or *phases*. Each phase performs a series of steps with *deliverables* associated with them.

Information Engineering Pyramid



The deliverables from one stage become the requirements for the next phase. Each phase in I.E. consists of a progressive refinement process, which takes a project from strategy, to logical, to the actually building and delivery and a system. Below are the stages of I.E.

Information Strategy Plan (ISP)

This phase maps out the IT strategy for an organization. Deliverables include a logical data model, logical process model, system model, and an IT Strategy Plan.

Business Area Analysis (BAA)

The BAA discerns major business areas of the organization from the deliverables of the ISP. These business areas may become one to many system development projects. The work performed in the BAA determines the *what's* of the business area. For example, what functions would a billing area perform? Deliverables include a completed logical data model, completed logical process model, state transition diagram, and CRUD Matrix diagram.

Business Systems Design (BSD)

The BSD further refines the logical requirements developed in the BAA from the *what's* to the *how's*. For example what functions are required for a successful implementation of a billing system? This phase starts the transformation from logical requirements to technical specifications. Deliverables include a physical data model, and a completed process model, along with completed process modules.

Technical Design (TD)

TD completes the transition from logical to physical. It maps the logical functions into technical development modules, used for coding traditional 3 GL's or Rapid Application Development (RAD) GUI Tools. The TD also converts the Physical data model's Entity-Relationship Model into a data schema that can be used for the creation of a database. Deliverables include process modules, data schema, and interface strategy.

Construction

This phase develops the process modules into back-end programs and user interfaces screens (normally GUI). All modules are first unit tested, and then the entire package is system tested. The database is created, and performance testing is carried out, to determine how best to tune the database for the new application. Construction concludes with a user/project review, to ensure that all requirements are met.

Curriculum to Information Engineering Matrix

As stated, each class will have sample requirements and sample deliverables that the students will process and use as a learning vehicle. By using the Information Engineering methodology, students will be exposed to real world requirements and deliverables. (see table below)

Class	Requirement	I.E. Stage	Deliverables
IS: Strategies & Management	Problem Statement	ISP	IT Strategic Plan High-level Entity-Relationship Diagram High-level Process Diagram
IS: Planning and Analysis	IT Strategic Plan	BAA	Enterprise Entity Relationship Diagram Enterprise Process Diagram Process/Entity Matrix Diagram
IS: Design and Construction	Enterprise E-R Diagram Enterprise Process Diagram Process/Entity Matrix Diagram	BSD	Process/Attribute Matrix Diagram Function Hierarchy Module Specifications
Introduction to CASE	Sample E-R & Process Models	TSD	Student developed E-R and Process Models
Introduction to LAN	IT Strategic Plan	TSD	Functional Local Area Network
Advanced Networking	IT Strategic Plan	Construction	Advanced Network Deliverables
Client Server Fundamentals I	Enterprise E-R Diagram	TSD	Completed E-R Diagram Entity-Attribute Report SQL-generated database
Client Server Fundamentals II	SQL-generated database Function Hierarchy Module Specifications	Construction	GUI-developed Client/Server Application
Introduction to Programming & Program Design	Function Hierarchy Module Specifications	TSD	Structured Program Specifications
Programming Lab	Structured Program Specifications	Construction	COBOL, Visual BASIC, or C/C++ developed Applications

Principles of Information Technology I **3 hours**

Prerequisites: *None*

Course Description:

Students will gain a thorough understanding of microcomputer hardware and software. Emphasis of the course is on learning applicable operating systems, using word-processing and spreadsheet software and how to apply these applications in a business environment. Students will also be introduced to style requirements for papers and projects.

Course Deliverables:

<Introduction course, outside model scope>

Principles of Information Technology II **3 hours**

Prerequisites: *Principles of Information Technology I*

Course Description:

Students will learn how to use multi-media presentation techniques in developing handouts, overheads, outlines, drawings, graphs, and professional presentations using presentation software. Database concepts are introduced, and students will design and build a database using database software.

Course Deliverables:

- <Introduction course, outside model scope>

Computer Based Research Fundamentals **3 hours**

Prerequisites: *Principles of Information Technology I & II*

Course Description:

Fundamentals of data creation, storage, retrieval, archiving, formatting, and reporting of information required in a business setting. Students will gain an understanding of various types of information and database systems - internal and external to organizations. Emphasis is on retrieval of relevant and timely information from appropriate sources for use in tactical and strategic organizational environments. Access to the Internet, and other appropriate on-line services, is an essential component of this course.

Course Deliverables:

- <Introduction course, outside model scope>

Introduction to Programming and Program Design
3 hours

Prerequisites: *Principles of Information Technology I & II*

Course Description:

Introduction to logic design and fundamentals of programming languages. Topics include an introduction to structured programming, algorithm design, control structures, pseudocode, modularization, communication between modules, cohesion and coupling, and general design criteria in solving business problems. All CIS students are required to take this course.

Course Deliverables:

- Structured Program Specifications

Programming Lab **3 hours**

Prerequisites: *Introduction to Programming and Program Design*

Course Description:

This is the laboratory portion of Introduction to Programming and Program Design. The course requires students to implement business systems in various computer languages. (Examples are COBOL, Visual BASIC, or C/C++). Students work on an independent basis and in teams to complete assignments.

Course Deliverables:

- COBOL, Visual BASIC, or C/C++ developed Applications

Client Server Fundamentals I **3 hours**

Prerequisites: *Principles of Information Technology I & II, Introduction to Programming and Program Design, Programming Lab*

Course Description:

This course introduces the student to the fundamentals of client/server computing. The course focuses on server based processes. Content includes database design concepts including the physical and logical organization of data, data relationships and operational requirements of database systems. The student will gain experience in designing a database structure using a relational database program *(Oracle)* on a microcomputer. Introduction to ANSI-SQL.

Course Deliverables:

- Fully documented Entity-Relationship Diagram and Entity-Attribute Report
- DDL (Data Definition Language) of all required database objects (tables, views, and sequences)
- PL/SQL-Based Report of Populated Database

Client Server Fundamentals II 3 hours

Prerequisites: *Client Server Fundamentals I*

Course Description:

This course concentrates on client processes and interactions in 2 and 3 tier structures. Introduction to client interface design using 4GL *(Powerbuilder, Oracle Forms)* to access databases. Introduction to RAD and JAD. Introduction to middleware, data access to heterogeneous databases, gateways to mainframe systems, transaction processing monitors, and communications and object brokering services.

Course Deliverables:

- GUI-developed Client/Server Application

Introduction to Local Area Networking 3 hours

Prerequisites: *Principles of Information Technology I & II*

Course Description:

Workstation and networking hardware fundamentals; design of local and wide area networks, installation of servers and clients. Networking Management tools. Emphasis on Novell NetWare and Microsoft Windows NT.

Course Deliverables:

- Functional Local Area Network

Advanced Networking 3 hours

Prerequisite: *Introduction to Local Area Networking*

Course Description:

Fundamentals of Networking Technologies. Various topologies and access methods. Network security; integration with other operating systems; fault tolerant systems; storage management and backup; troubleshooting principles. Preparation for certification examinations.

Course Deliverables:

- Advanced Network Deliverables

Introduction to Computer Aided Software Engineering 3 hours

Prerequisite: *Introduction to Programming and Program Design*

Course Description:

In this course the student will be introduced to structured analysis and design techniques using a CASE tool. The student will learn the tools of structured analysis and design, including data flow diagrams, entity relationship diagrams, context diagrams, event lists, data dictionary, process specifications, state transition diagrams, and structure charts.

Course Deliverables:

- Student developed E-R and Process Models

Information Systems: Planning and Analysis 3 hours

Prerequisite: *Client Server Fundamentals I, Advanced Networking, Computer Aided Software Engineering, Introduction to Programming and Program Design*

Course Description:

An in-depth study of the first three phases of the System Life Cycle: Preliminary Investigation, Analysis, and Design. The student will develop an enterprise business model, enterprise data model, and develop an Information Technology Implementation Plan. The second phase will emphasize analyzing the current situation: organization, data, and functions. Requirements for organization, data, and function will be developed. Design will concentrate on overall system structure and flow, organization structure and flow, selection of computing, communications, and data platform, and design of application programs, documentation and training materials. Object Oriented Analysis and Design will be introduced. Heavy emphasis on modeling using CASE tools. Introduction to Project Management techniques. Introduction to ethical behavior required of Information Systems Professionals. (DPMA model to be used as reference)

Course Deliverables:

- Enterprise Entity Relationship Diagram
- Enterprise Process Diagram
- Process/Entity Matrix Diagram

Information Systems: Design and Construction
3 hours

Prerequisites: *Information Systems: Planning and Analysis*

Course Description:

This course covers the Programming and Installation phases of the System Life Cycle. During this course students will program, test, and install an information systems project involving multiple programs. Programming may use any of the tools previously learned (COBOL, C, SQL, Powerbuilder, etc.) Alternative development methods are investigated (RAD, Prototype, JAD, etc.) Continued emphasis on using CASE and other productivity tools. Project Management techniques will be used throughout the development process. Continued emphasis on ethics.

Course Deliverables:

- Process/Attribute Matrix Diagram
- Function Hierarchy Module Specifications

Information Systems: Strategies and Management
3 hours

Prerequisite: *Information Systems: Planning and Analysis*

Course Description:

This course begins with a review of organizational strategic plans and their scope and associated implementation activities. This provides the framework for an investigation of the methods appropriate for IS planning and management. Elements of IS planning covered: facilities, equipment, personnel, technical skills and specialties, training, replacement of equipment, interdepartmental collaboration and prioritization, and management information requirements. Current issues in ethical standards and behavior are included. Introduction to Business Process Reengineering. The course is a capstone experience for CIS majors. All elements of previous courses will be incorporated with heavy emphasis on the management techniques in a global business environment.

Course Deliverables:

- Information Technology Strategic Plan
- High-level Enterprise Entity-Relationship Diagram
- High-level Enterprise Process Diagram

Directed Study in Information Systems **1-3 hours**

Prerequisite: Permission of the Program Chairperson

Course Description:

Students wishing to investigate a topic not otherwise studied in their curriculum may elect a one (1) to three (3) semester credit hour directed study course. The directed study will earn general elective credit exclusively and may not be substituted for a required course (including required elective course work). A directed study request form must be completed by the student and approved by the Registrar and the department chair supervising the study. The form may be obtained in the Records Office.

Course Deliverables:

- Instructor's Discretion

INFORMATION SYSTEMS: DEVELOPING A WRITING ACROSS THE CURRICULUM EXPERIENCE

Marian Sackson
Information Systems Department
Pace University

ABSTRACT

A student's ability to articulate thoughts through writing is an important component of the total educational experience. Past scores on the SAT verbal scores emphasized the need to improve student's ability to express thoughts through the written word. Thus, the author undertook a *writing across the curriculum experience* to strengthen the student's awareness of the interdisciplinary nature of writing. The vehicle used was a journal of assigned reading of computer-related periodicals. Over time, after extensive writing critiques, the journal entries of the students improved. The students did not always enjoy the additional curriculum assignment, but they arose to the occasion.

Introduction

Most students at the college and university educational levels need to improve their combination of thinking and writing skills. One approach to confronting this dilemma is *writing across the curriculum.* A student's ability to articulate thoughts through writing is an important component of the total educational experience. However, many disciplines within the educational process do not emphasize or reinforce the thinking and writing components.

We are a large private university in both a major urban and suburban Northeast setting. Our student body is predominantly first or second generation college attendees, approximately 25% are minorities, and they come from both public and parochial high schools. The fall 1995 freshmen class was 1561 in number, with 859 mainstream students, 78 honors students and 460 special-needs students.

The average verbal Scholastic Aptitude Test (SAT) score at our university for Fall 1995 entering freshmen was 435, with a subgroup high of 700 and a low of 325. Performance on this objective test supports the need for a broader based focus on reading, writing, and comprehension skills. Imbedding writing across the curriculum reinforces the application of writing skills in all disciplines. The accountant must write reports, the secretary must know grammar, the scientist must write logically, the teacher must write clearly, and the actor must comprehend his lines.

The author chose the goal of writing improvements. To this end, the author introduced a written component into a college level introductory computer systems concepts course (IS241). The objective was to encourage and support the successful application of word processing programs as tools to aid the students' writing skills.

Needs Assessment

The results of the recent administration of the verbal component of the national SAT scores alerted the educational community to a decrease in verbal test performance. This fact reinforced the need to develop ways to improve the thinking and writing component of a student's educational experience.

One might never find the perfect paradigm for the complex relationship between thinking and writing, but many educators believe there is a connection. Students develop writing skills slowly (Valvoord, 1982, pp. 12-16). "A body of research in psychology indicates that when people verbalize an idea, they can remember it more thoroughly, and manipulate it in more sophisticated ways. Writing is one of the most effective modes for learning, Because writing and thinking develop in complex ways across a student's years of schooling, every instructor needs to support their development" (Valvoord, 1982 pp. 3-5). Thus, teaching students to write well involves teachers in various disciplines as well as those who teach writing classes.

Students might object to this emphasis on the development of writing skills. These negative reactions include:

1. I use a computer for word processing to check my spelling and grammar.
2. The current communication technology, such as E-mail, does not require me to do extensive writing.

A response to these comments could be that business decision making leads to communicating with someone to put the decision into effect. Likewise, the thinking-communications process often requires business people to convey their thoughts through written words (Lawrence, 1974, pp. 25). In addition, many middle and upper level managers who lecture and participate in the business schools of numerous colleges and universities express their concerns. They believe that graduates, who will become the future employees, need to have the ability to write clearly, legibly, and in a style appropriate to the subject of the document.

Students show a greater command of a subject about which they had written assignments rather than only reading or listening to a classroom lecture (Weiss & Walters, 1979, pp. 6-8). Writing is a uniquely effective tool for learning. Writing and learning are many faceted. Thus, learning serves an analytical and connective function, as does writing. This experience organizes individual facts, images, and symbols into sentences and essays. Learning at its best is engaged, committed, and self-rhythmed, as is the best writing (Emig, 1977, pp. 123).

"Look at how we teach people to think. We have teachers stand in front of classrooms and think out loud while students listen ...Can we do any better?" (Kugel, 1988, pp. A27). The writer alleges that writing across the curriculum suggests that to make students think one must make them write about it. "Those who advocate writing across the curriculum are suggesting a different kind of writing: writing to learn, writing to develop ideas; writing to help you think" (Kugel, 1988, pp. A27).

At our university all entering freshmen are required to take a writing skills test. As a result, many students need to participate in a special writing course to compensate for their writing deficiencies. Two hundred and twenty-four, or 57% of the mainstream entering freshman class are placed into Reading 100. It is a remedial course devoted to developing effective writing skills through review of basic grammar, sentence structure, and usage.

Due to these numbers of students attending the special writing classes, the academic administrators of our university realized there are serious problems with students. The students are unable to articulate their thoughts through writing. In recognition of this problem, our university offered a Writing Across the Curriculum seminar to all interested faculty members.

Even though writing deficiencies are discovered through tests administered to entering freshmen, writing across the curriculum can occur at any period in a student's college experience. Hence, the author decided to require extensive writing in the Information Systems Concepts course. This course teaches the students concepts of information systems analysis and design. A system analyst applies new technologies to business and industrial problems. A perquisite to these 'technology transfers' requires the analyst to keep abreast of the latest trends and techniques. The best way to accomplish this is to develop a disciplined reading program. It is here that the author of this paper decided to combine the reading program with writing assignments to strengthen students' awareness of the interdisciplinary nature of writing.

Objectives

Students will derive writing benefits through (1) keeping a journal in which they record their reactions and experiences to reading articles in computer-oriented periodicals and (2) becoming accustomed to reading these periodicals. The journal writing will help the students' develop the habit of reading, reflect on the reading and become used to the feeling of words pouring out onto a paper.

Twenty-two to twenty-eight students enroll in the Information Systems course. They are all familiar with word processing programs and use them either in the academic computer lab or at home.

The benefits of exposure to the writing across the curriculum experience are:

1. Student recognition of the need for writing skills in a non-writing course.

2. Exposure to computer-oriented periodicals related to information systems.

3. The use of a word processing program for revision, editing and proofreading before submitting their journals.

Activities

Two projects are assigned to students as follows:

1. They are given a list of computer-related periodicals maintained in the university library. They are told to initially browse through periodicals since they will not have time to read everything. The browsing assignment is as follows:

    a. They will browse through periodicals about one or two hours per week.
    b. They will study the table of contents of ***two*** periodicals. They will read the first paragraph, conclusion or last paragraph of any two articles in each periodical that seem interesting.
    c. They will record their browsings. This will include, along with their name and submission date, the titles of the four articles, and the name and date of the periodicals. In addition, they will also write the first paragraph, summary or conclusion of one of the browsings. They will submit the ***browsings*** using a word processing program.

2. They are to select ***one*** of the four articles that they browsed and read it thoroughly for their journal. The journal assignment is as follows:

    a. Write a brief summary of the article in one or two paragraphs.
    b. Write your evaluation of the article.
    c. Write your personal evaluation of the periodical itself (this could be either positive or negative.)
    d. The journal should contain ***three*** double spaced pages, unjustified and in twelve point text.

The journal assignment, given to each student, illustrated documentation using the APA style. The format of the journal required each student to write their name, due date, and references in the APA style that included name of the author, periodical and date of the periodical.

Each journal entry was evaluated on (1) the quality of the writing and (2) the content of the journal entry. The first journal entry was due one week after the initial assignment. This brief turnaround from assignment to performance had a specific purpose. The author intended to quickly evaluate the initial journal for the writing and verbal articulation skills. If students were able to know quickly that they had writing deficiencies and what they were, they could endeavor to improve the subsequent journals.

They were assigned a total of six journal entries and twenty-four browsing entries. One journal entry was due every two weeks along with four browsings. The students were required to keep their journal entries in a folder until the end of the semester. Then they resubmitted them. To ensure the seriousness of this writing across the curriculum experience, the complete journal and browsing entries were worth ***thirty*** percent of the final grade.

Conclusion

There were improvements in the writing skills of the students as the semester progressed. The initial journal submissions illustrated that some students had thought and writing deficiencies. Comments on these journals alerted these students to spelling errors, incomplete, run-on or poorly constructed sentences as well as incoherent paragraphs. Those students that had major writing deficiencies were referred to the writing center.

The improvements in the journal grades portrayed that (1) students who had minor writing deficiencies quickly corrected their problems and produced well written writing assignments and (2) students with definite writing problems consciously tried to improve their writing skills. The students seemed to become aware that reading and writing are significant parts of a computer-related career. The allocation of thirty percent of the final grade to the journals also added a driving force to the importance of writing well.

The journal exercise was a successful Writing Across the Curriculum experience. However, some of the students did not totally appreciate the assignment. Many felt that, along with other assignments and tests required for the Information Systems Idea course, this was a substantial additional requirement for one coarse. Numerous students combine working with school attendance. Thus, any additional coarse requirements become important to their balance between working and school related responsibilities.

The author hopes that this discussion of Writing Across the Curriculum illustrates the importance of the topic. Furthermore, the author chose the writing method that was appropriate to the particular coarse curriculum. Therefore, any interested faculty should choose their own pertinent experience when implementing Writing Across the Curriculum.

REFERENCES

Buscemi, Santi. (1990). A Reader for Developing Writers. New York: McGraw-Hill.

Emig, Janet. (1997). Writing as a Mode of Learning. College Composition and Communication. 28, 122-28.

Fiske, Edward. (1990, January). But Is the Child Learning? Schools Trying New Tests. The New York Times.

Flower, Linda. (1981). Problem-Solving Strategies for Writing. New York: Harcourt.

Gagne, Robert and Ernest Smith.(1965). A Study of the Effects of Verbalization on Problem Solving. Readings in the Psychology of Cognition. ed. Richard C Anderson, New York: Holt.

Kugel, Peter. (1988, October 24). Why Johnny Can't Think. The New York Times.

Lawrence, Nelda. (1974). Writing Communications in Business and Industry. 2nd ed. New Jersey: Prentice-Hall.

Schwartz, Mimi. (1985). Writing for Many Roles. Portsmouth: Boyton/Cook Publishers.

Turner, Judith. (1990, December). Use of Computers Can Improve Students' Writing Ability, Study Shows. Chronicle of Higher Education.

Troyka, Lynn. (1990 2nd ed.). Simon and Schuster Handbook for Writers. New Jersey: Prentice Hall.

Valvoord, Barbara, (1982). Helping Students Write Well: A Guide for Teachers in All Disciplines. New York: Modern Language Association of America.

Weiss, Robert and S. A. Walters. (1979). Research on Writing and Learning: Some Effects of Learning-Centered Writing in Five Subject Areas. Session on Research in Composition, National Council of Teachers of English Convention. San Francisco.

Woodford, Peter. (1969) Sounder Thinking Through Clearer Writing. Science, 745.

Zinsser, William. (1988). Writing to Learn. New York: Harper and Row.

COOPERATIVE EDUCATION: A DIFFERENT PERSPECTIVE

Gerald E. Wagner, Ed.D., CCP, CDP, CISA, CDE
Professor Emeritus, Computer Information Systems
California State Polytechnic University, Pomona
Phone: (909) 869-3238
FAX: (909) 869-3248
EMAIL: gewagner@csupomona.edu

Doris G. Duncan, Ph.D., CCP, CDP, CSP, CDE
Professor Computer Information Systems
California State University, Hayward
Phone: (510) 885-3364
FAX: (510) 885-4796
EMAIL: dduncan@csuhayward.edu

ABSTRACT: This paper describes the problems facing Information Systems (IS) educators in keeping curricula current and producing graduates with skills that are valuable to employers. These problems are often magnified by rapid changes in technology. This paper proposes solutions to these problems through the creation of cooperative learning alliances with industry. The first section of this paper summarizes the problems facing IS educators. Then definitions and unique examples of cooperative education and its benefits are explored.

INTRODUCTION

The many advancements in computer technology in the last few years, both hardware and software, have brought new challenges and opportunities to industry and academia alike. As most computer professionals are aware, the useful life of computer hardware was measured in years during the 1950s, '60s, and even into the '70s; but we now find new models of computers being introduced almost every month. Systems design methodologies and software development tools are also changing rapidly. "Because change is one of the few constants in technology, the business/academic connections must be revisited often" (1).

THE CHALLENGES

In an attempt to remain competitive, companies and organizations that rely on computers are being forced to continually upgrade their systems. This is not only costly, but in a period of time when the emphasis has been on corporate downsizing (right sizing), mergers, etc., it is extremely difficult to obtain and/or keep personnel who are knowledgeable in the use of the latest computer technology. For example, "We see a multitude of new job titles in this arena that did not exist ten years ago (e.g., database analysts, telecommunications/networking specialists, EDP security/auditing specialists, user support analysts, and office automation specialists)" (2).

In addition to the problems which are caused by the rapidly changing technology, the budgetary constraints under which many MIS Departments are working makes it very difficult, if not impossible, to provide the training needed to keep personnel up-to-date. On the other hand, it has long been considered the responsibility of our educational system to provide graduates with the background and skills necessary to be successful in their chosen fields of employment. For that reason, when employers do recruit graduates, they look to colleges and universities with a curriculum that utilizes new technology and emphasizes current practices (3). Because of the rapidity in which technological changes are taking place, computer educators are finding many obstacles to fulfilling this responsibility. These obstacles include limited budgets, lengthy curriculum review cycles, obsolescence of hardware, software, and faculty skills, proliferation of new course topics, and the ever changing demands of industry. Historically, the curriculum requirements for a degree that were in effect at the time a student entered college are the same requirements that must be met four, five, or even more years later when the student is ready to graduate. Although this rule was initially designed to protect students who might never graduate due to ever changing requirements, it also means that students who are majoring in a technology-oriented discipline may be graduating with skills that are obsolete by the time they complete the stated requirements for graduation.

To compound the problem, many universities have a two- to three- year curriculum revision process. Also, many universities attempt to save money by printing new catalogs once every two years. This means that a new course, or one with major content changes, will not be approved and printed in the catalog until two or three years after initially proposed. With the requirement that a course be included in the catalog prior to being scheduled, many courses are already out-of-date before they can be taught for the first time. "Future curriculum planning must be an on-going activity" (4). Assuming that this is true, some changes must be made in the present curriculum revision process. One solution to the problems associated with a lengthy curriculum review and approval process is the use of non-specific course titles and descriptions. For example, rather than using the title, "C++" or "Smalltalk", a title such as "Object-Oriented Programming" could be used with any object-based language. By being less specific, the tool could be changed to conform to the needs of primary employers of graduates without necessitating a change in course title. The course content could also be changed as needed, if a generic, problem-solving-oriented course description were used. Presently, too many course descriptions are overly specific, requiring that they be changed annually and thereby compounding the problem of obsolescence.

Another major problem faced by colleges and universities is a problem similar to that faced by industry: keeping instructors up-to-date and qualified to teach students how to use the latest computer hardware and software. Even if instructors do find ways to keep current personally and then attempt to modify course content to include the use of new technology, they often find it necessary to develop their own teaching materials because there are no up-to-date instructional materials.

Based on the experience of the authors, the problem most frequently mentioned by educators and students is the lack of hardware and software representing the latest technology. Some academic institutions are attempting to solve this problem by requiring students to provide their own personal computers and related software. This results in a shift of the budgetary problem from the institution to the student and, at best, is only a temporary solution because the institution is obliged to, but doesn't always, provide appropriate hardware and software to the faculty for research and course development.

So far numerous problems have been identified, but few solutions have been offered. In fact, the authors believe that the above problems will only become worse if educational institutions and industries do not work together to solve these problems. In our case, organizations need employees who are skilled in the use of the latest computer technology, but they don't have the time or trainers to complete the training. Academic institutions, on the other hand, have faculty with the expertise to provide the training needed by industry, but are unable to acquire the latest versions of hardware and software. "Some researchers have concluded that businesses and universities share some similar challenges, and increased cooperation between the two entities will assist in shared solutions for both" (5,6). The authors believe that the key word is "cooperation." By combining their resources and working together cooperatively, many of the problems and shortcomings faced by each can be alleviated. Based on our experiences, we believe that it is time for universities to join hands with industry and utilize cooperative education to its fullest.

COOPERATIVE EDUCATION PRESENTS OPPORTUNITIES

"The term 'cooperative education' has traditionally been used to describe work/study programs where a student alternates between working full-time and attending classes full-time" (7). Although this is still a valuable form of cooperation, the authors believe that the spirit of cooperation between businesses and institutions of higher learning must be expanded and taken to a new level. Some of the ways in which this can be done are summarized in Table 1 and discussed in the following sections of this paper.

TABLE ONE
COOPERATIVE EDUCATION METHODS

1. Advisory Committees
2. Student Internships
3. Industry Projects for College Seniors
4. Faculty Internships
5. Faculty/ Industry Exchange Programs
6. Faculty Training and Updating
7. Hardware/Software Resource Sharing

Advisory Committees. A problem that is of particular concern to CIS/MIS educators is that of attempting to keep the curriculum up-to-date. Which software packages, programming languages, systems methodologies, and case tools should be taught and/or utilized? What hardware should be used? More often than not, there is no single answer to these questions. Each campus must determine the needs of the community it serves. That is, what are the needs of the major recruiters or companies in the area?

Rather than work in a vacuum, universities should determine what computer hardware, software, and applications are being used in the organizations which hire their graduates. An excellent way to accomplish this is through the formation of an advisory committee which is made up of alumni, MIS Managers, and recruiters. The individuals selected should be in a position to know the needs of their organizations, as well as having a feel for the direction that the computer industry is moving. Traditionally, alumni of the program and managers who have hired graduates of the IS program are in a position to provide input regarding the relevancy of the content of individual courses and the curriculum as a whole.

It is recommended that this advisory committee meet as a body at least once each year. Preferably, the meeting would be scheduled one or two months before the deadline for submitting curriculum change proposals. The agenda for each meeting should include a report on hardware and software in use or on order by the university, curriculum changes being considered, and any problems since the last meeting of the committee. Faculty requests for resources and curriculum changes that have been recommended or supported by an industry advisory committee are usually looked upon more favorably by the campus administration than those submitted by departments that do not utilize advisory committees.

Student Internships. The term internship is used to describe all programs where students work part-time in a position related to their chosen career field while continuing to make progress toward their degree objective. Student internships are not new, but they are often not utilized by industry. The question might be, why aren't student internships used more by businesses considering the many benefits that can be derived? Some of the benefits to the employing agency are: (1) a ready source of "skilled" applicants, (2) an opportunity to test employees without a long-term commitment, (3) a relatively inexpensive source of labor, (4) highly motivated employees, and in some cases, (5) applicants who have been "prescreened" by the university. According to a 1994 New York Times article, "Employers nationwide are hiring four of every five of their college interns full-time after graduation" (8).

The benefits are not limited to employers. Internships also provide benefits to the university. An internship provides (1) a real world test of the academic program, (2) a supplement to classroom learning, (3) access by the student to expensive resources, (4) a ready market for the product (graduate) of the university, and (5) a motivated student because classroom learning becomes more relevant.

Industry Projects for College Seniors. Complex subjects are learned best by students if the students are involved in a variety of different teaching methods. Students can read about the subject, observe demonstrations, listen to lectures by an instructor, or be actively involved in the learning process through the assignment of a project or case study. It has been proven that students learn best by physically doing something. In this case, the more realistic the project or case study assignment is, the more valuable the learning.

Since most companies have a backlog of systems and programming projects, it would seem quite likely that one or more of those projects could provide a valuable learning experience to students. A key consideration in selecting real projects from industry is that the projects must: (1) fit the timeline of the term, either 10 or 16 weeks, (2) have a required deliverable at the completion of the project, and (3) provide a learning experience consistent with the learning objectives of the course and curriculum.

Real life senior projects from industry provide a good introduction to the problems that will be faced after graduation. Of particular value is the students' realization that communication skills are important if they must work in a team environment and meet periodically with the industry liaison. The university benefits inasmuch as many of the projects involve hardware and/or software that is not available at the institution. The cooperating organization gains by having a low priority project completed at virtually no cost.

Faculty Internships. Faculty internships, unlike student internships, usually involve full-time employment during the summer. In some instances, a faculty internship could require the faculty member to take an unpaid leave of absence from teaching for one or more terms. Regardless of the duration of the internship, the employer and university both enjoy benefits from the program. The employer gains from the expertise of the faculty member, while the faculty member gains insight into the use and application of some of the latest computer technology.

Faculty/Industry Exchange Programs. A few large corporations, such as International Business Machines, have been involved in faculty/ industry exchange programs for many years. In this program, a faculty member is assigned to work for the cooperating organization while an employee of that organization is scheduled to work at the university. That work might consist of teaching courses, assisting in new course development, and/or training other faculty in the use new hardware or software. These exchanges are usually of a short duration, usually one year or less, and the salary and fringe benefits of the people involved are maintained by the original employing agency.

Faculty Training/Upgrading. One of the problems faced by both businesses and academic institutions is that of attempting to keep personnel up-to-date with the latest technology. Most large organizations attempt to overcome this problem by providing training for their employees. In many instances, this involves paying all expenses to send one or more employees to another training site for a few days--in some cases one or more weeks. This is extremely expensive if many employees must be provided with the same training. In those cases, organizations often hire a professional trainer and hold the training sessions at the local place of business. When this is done, the cost of the training is usually a fixed amount regardless of the number of individuals being trained. Therefore, it costs the company nothing to invite one or two faculty from the local university to attend the training session. The only cost to the university is that of providing a substitute teacher for the classes missed and, in many CIS/MIS Departments, the missed classes are taught as an overload by other faculty in the department.

Hardware/Software Resource Sharing. In the past, many companies have donated computer equipment that they no longer need to academic institutions. Although this is still a possibility, there is a high probability that any equipment being offered as a donation is already obsolete and will not be of any major benefit to an academic institution either. Another way in which access to needed hardware/software can be provided to universities by industries is via networks. Since most companies are now using networks to communicate with different entities within their organization, the CIS/MIS Department at the university can be established as a node on the company's network. It may be necessary to limit access by the university to certain hours of the day so that the ongoing operation of the company is not impacted. In addition, care must be taken by the company to assure that access is controlled and limited to only those areas previously agreed upon. With proper controls, valuable resources can be shared with an academic institution with virtually no extra cost to the company.

Some much needed software cannot be obtained by universities because of the high cost of licensing. On the other hand, it is quite possible that a local company may have a licensing arrangement that provides more copies of software than that company actually needs. For example, a company that needs 30 copies of a software package may find that the cost is the same for 25-50 copies. Some of the excess copies, with permission of the software company, can be "loaned" to the university. In the long run, the company benefits because graduates will be trained on the software that companies are using.

As noted above, the sharing of software or use of software licensed to another person or entity should only be done with the written permission of the software vendor. Based on the authors' experience, such permission can usually be obtained if the software vendor is involved early in the decision process and knows how and where the software will be utilized in the curriculum..

SUMMARY

The above suggestions are only some of the ways in which organizations and universities can work cooperatively to overcome some of their problems and achieve their objectives. It should be noted that the benefits that can be attained far exceed any costs that may be involved in the implementation of these suggestions. Many of the suggested areas for cooperation presented in this paper have been proven effective at both of the authors' universities, and we are of the opinion that all of the suggestions for cooperative education could work to some degree for all colleges and universities.

REFERENCES

1. Heiat, Heiat, and Spicer, "Future Software Training Needs: Contrast in Needs as Perceived by Business and Academia", Journal of Information Systems Education, Fall, 1995, pp. 102.

2. Mawhinney, Charles H., Morrell, Joseph S., and Morris, Gerard J., "The Undergraduate Curriculum: Closing the Gap", ISECON '94 Conference Proceedings, Park Ridge, IL.: Data Processing Management Association, pp. 249-253.

3. Davis, L., "Can Education Meet IS Career Demands?", DATAMATION, March 15, 1989, pp. 65-69.

4. Kruk, Leonard B., "Information Systems Integration: A Future Forecast", The Thirteenth Annual Office Systems Research Conference Proceedings (March 27-29, 1994), pp. 188-195.

5. Wright, Phillip C., "The CEO and the Business School: Potential for Increased Cooperation?", Journal of Education for Business (September/October, 1992), pp. 32-37.

6. Christensen, Anne L., and Philbrick, Donna, "Business and Universities: Similar Challenges, Shared Solutions", Journal of Education for Business (September/October, 1992), pp. 6-10.

7. Wagner, Gerald E., "Internships: A Bridge Between Theory and Practice", ISECON'95 Conference Proceedings, Park Ridge, IL.: Data Processing Management Association, pp. 187-193.

8. Lum, Lydia, (New York Times Service), "Firms Hiring 4 Out of 5 of Their Interns", Inland Valley Daily Bulletin, January 30, 1994, pp. A13.

It's Time to Apply Good MIS Principles to Higher Education's Use of Information Technology

by
Kathy Schwalbe, Asst. Professor and MIS Area Coordinator
Augsburg College, Minneapolis, MN (schwalbe@augsburg.edu)

Abstract

Although colleges and universities have increased spending on Information Technology (IT) dramatically over the past decade, there is little formal evidence linking it to higher productivity or changes in the educational process. Are colleges and universities using good MIS principles and focusing on providing needed services and information resources to students and society, or are they focusing on the technology itself?

Analysis of data from 108 national colleges and universities' use of twelve different technologies supports the following propositions:

- There is a relationship between the quality of higher education institutions and their use of particular information technologies. In general, most technologies are used more by higher tier schools, except for distance learning, which is used more by lower tier schools.
- The more a technology is used in an organization, the more effective it is perceived to be. This was true in all cases where data was provided. However, none of the eight technologies used in this study were perceived as being highly effective in improving the teaching and learning process.
- There are differences in motivations for using particular technologies. Having a commitment to staying on the leading edge of instructional use of technology was the number one motivator for using all of the technologies except e-mail, multimedia, and distance learning, which were the only technologies which listed being a part of the school's mission as a top motivator.
- Colleges and universities are generally not focusing on pedagogical or productivity-related reasons for adopting information technologies. Multimedia technology was the only one where clearly pedagogical reasons made the top five list (out of twenty-two) motivators. Reducing the overall cost of instruction over time never made the top five list of motivators.

Introduction

There have been many studies and books written about the use of IT in industry, and numerous theories and models concerning IT usage and management have been developed. Several companies have shown dramatic increases in performance by using IT to transform or reengineer their business processes. American colleges and universities have also increased their use of IT dramatically over the past decade, but few studies have addressed the details behind this movement. Many groups question the management of IT resources in higher education as costs continue to rise and outcomes remain in question.

This investigation begins to analyze the use of different types of information technology in higher education, reasons for their use, and effectiveness of their use. Relationships between high level institutional outcomes and the use of different technologies are also examined. Comparisons of what higher education is doing and what industry has learned are also made.

The sample included in this study includes 108 national colleges and universities offering business programs. High level institutional outcome data was extracted from the U.S. News and World Report's 1994 College Guide, and detailed data about use of twelve different technologies was provided by UCLA's Anderson School of Business 1994 survey on business school computer usage. This article is based on a Ph.D. dissertation available in its entirety from the author.

Problem Statement

Some groups in higher education argue that it is long overdue for major reengineering or transformation. A 1993 report by the Wingspread Group on Higher Education concludes that colleges and universities must change quickly and dramatically, not only by educating more people but educating them better and cheaper. Many corporations have learned that information technology can be a key enabler to major transformations or business reengineering.

Institutions of higher education are very labor intensive. For many institutions, eighty percent or more of the operations budget is allocated to personal services. For at least a decade, the cost of personal services has been rising at about eight percent per year. The consequence has been tuition increases that have about doubled the rise in the Consumer Price Index in the past ten years and even outpaced increases in health care costs.

Some experts predict tuition costs to double between 1990 and 1997 at many public institutions, even though student-teacher ratios creep upward (Heterick, 1994).

In addition to increasing labor costs, higher education institutions have also spent almost $70 billion for computer-related goods and services in the last decade and a half (Green and Eastman, 1994). In 1994 alone, American colleges and universities spent more than $6 billion for information technology goods and services, of which roughly $1.75 billion will go to support the instruction mission. Given about 15 million in enrollments for fall 1994, this translates into annual spending of about $400 per student in overall technology and about $115 per student in educational technology support. But despite massive technology expenditures in the last decade or so and great advancements in information technology's power, users' comfort levels with the technology, and reduced unit costs, "information technology is not being integrated into the teaching and learning process nearly as much as people have regularly predicted since it arrived on the education scene three or four decades ago" (Geoghegan, 1994, p. 14). Some estimates suggest that information technology may be integrated into no more than five percent of the courses being taught in colleges and universities(Willut, 1994). Even though this number has increased in the last two years, there still is no evidence to suggest that IT is being used to help educate more students better and cheaper.

As the cost of higher education continues to increase, society's confidence in higher education continues to deteriorate. Numerous reports cite trends that call for changes in the way the academy operates. Politicians, the public, and the customers of higher education are demanding more than ever before. They want measurable evidence of improvements in quality of teaching and accountability for funds. However, "discussions of efficiency are both unfamiliar and uncomfortable to the academic world. Have you noticed that when the issue of efficiency is raised in a group of educators, the conversation immediately turns to discussions of quality as if the two were inextricably opposed to one another? We tend to assume that increasing academic productivity is good for taxpayers and for others who foot the bills, but perhaps not so good for students and faculty" (Twig, 1993, p. 13).

There appears to be an obvious gap in communications and intentions between people in higher education and many of its constituents. The intended outcomes of colleges and universities are in question, as are its management practices, particularly its costs. Since spending on IT and apparent lack of benefits are in the public eye, this topic is a critical one to leaders in higher education. It's time to apply good MIS principles to higher education's use of information technology.

Propositions Examined

This investigation examines whether colleges and universities have been using their investments in information technology to transform or reengineer the teaching and learning process to educate more students better and cheaper. It summarizes what types of technologies different tier institutions (based on the four tier rankings from the U.S. News and World Report's 1994 College Guide) are using and why they are using them. The following propositions are examined.

1. There are differences between the quality of higher education institutions and their use of particular information technologies.
2. The more a technology is used in an organization, the more effective it is perceived to be.
3. There are differences in motivations for using particular technologies.
4. Colleges and universities are generally not focusing on pedagogical or productivity-related reasons for adopting information technologies.

Summary of Findings

1. There are differences between the quality of higher education institutions and their use of particular information technologies. In general, most technologies are used more by higher tier schools (based on four tiers or quartile rankings from U.S. News and World Report), except for distance learning, which is used more by lower tier schools. Figure 1 shows the technology phase of adoption curve used in the UCLA survey on learning technologies (Frand and Ng, 1994). A phase of adoption of 6 means over 75% of the population uses it; 5 means 50-75% of the population uses it; 4 means 25-50%; 3 means no more than 25%; 2 means initial actions are being taken to use the technology; 1 means information is being gathered; and 0 means not applicable or no interest/support for the technology.

Figure 2 clearly shows that higher tier schools tend to be further along in using most technologies, except distance learning. This figure charts the mean phase of adoption for twelve different learning technologies by tier ranking of the national colleges and universities in the sample. A multiple regressions analysis of the phases of adoption for all of the technologies versus tier resulted in a significant R-square of .43. In other words, knowing the phase of adoption for these technologies reduces 43% of the uncertainty in predicting the tier of any institution, given no other information. In particular, results of this model show that distance learning and student laptop ownership have the most significance in predicting tier of an institution. This has definite equity implications if one assumes it is better to receive "live" instruction (Stoll, 1994) and that students can be more productive if they have easy access to computers (Astin, 1992, Krueger, 1992).

Figure 1. Technology Adoption Phases (Frand & Ng, 1994)

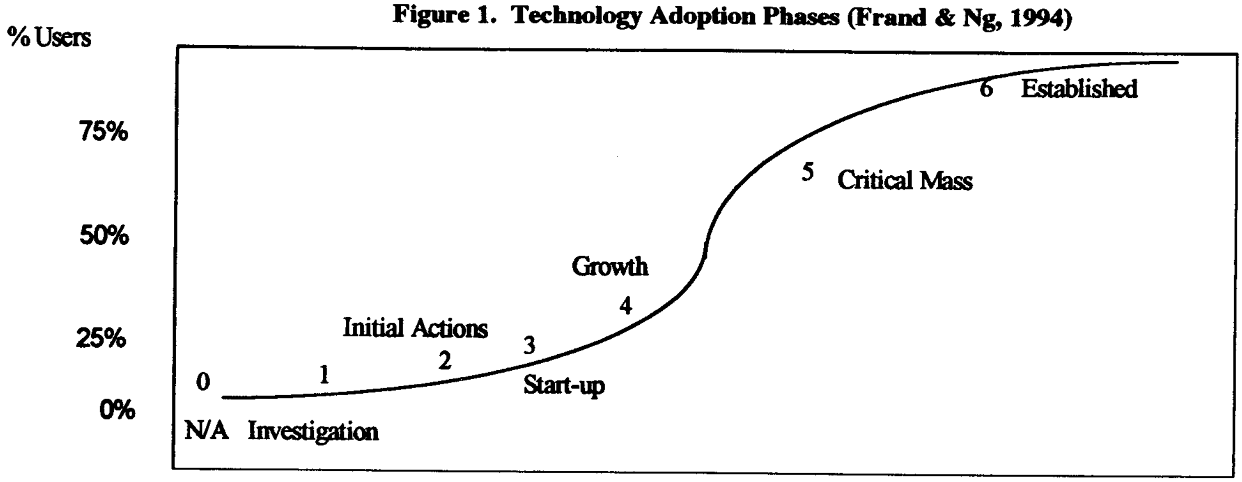
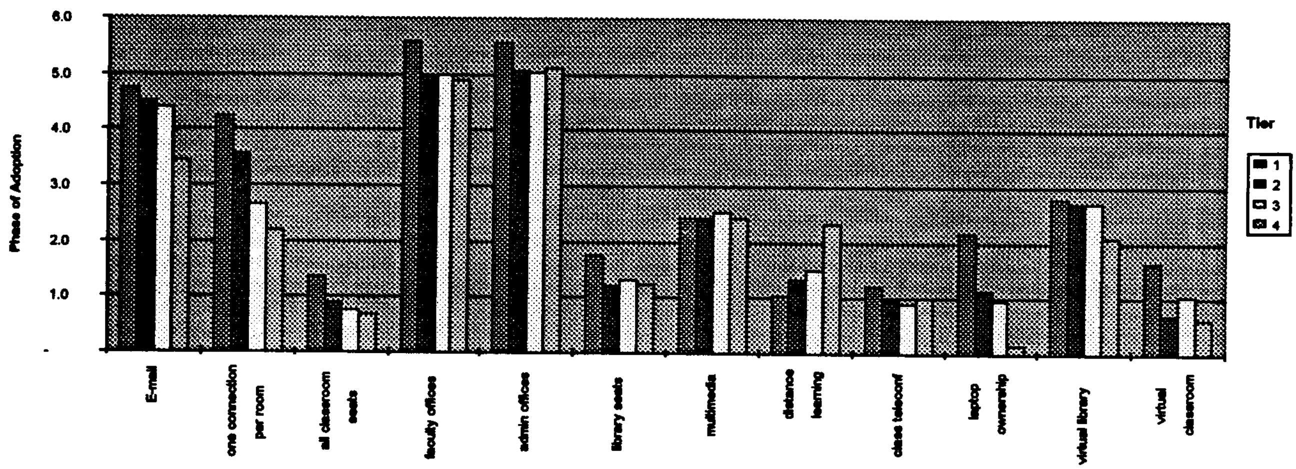


Figure 2. Tier of Institution Versus Mean Phase of Adoption of Technologies



2. The more a technology is used in an organization, the more effective it is perceived to be. With many innovations, benefits are often not realized until a certain number of people have adopted it. This was true in all cases where data was provided in the UCLA survey. Figure 3 summarizes the mean phase of adoption and mean effectiveness score for using the related technologies.

Effectiveness was measured on a five point scale (1 = not effective and 5 = very effective). The only technology that received a mean effectiveness rating more toward very effective than not effective was e-mail. Note, however, that the effectiveness question for e-mail was more general and related to "facilitating business school communication" while all of the other effectiveness questions specifically addressed the teaching and learning process. None of the other technologies had very high ratings for improving the teaching and learning process.

3. There are differences in motivations for using particular technologies. Figure 4 summarizes the top five motivators as well as the top five impediments for using different technologies in this sample. Having a commitment to staying on the leading edge of instructional use of technology was the number one motivator (out of a list of twenty-two) for using all of the technologies except e-mail, multimedia, and distance learning, which were the only technologies which listed being a part of the school's mission as a top motivator.

Figure 3. Mean Phase of Adoption and Effectiveness Scores of Technologies

Technology	mean phase of adoption	mean effectiveness score
e-mail	4.37	3.44
multimedia (enhancing classroom instruction)	2.45	2.90
multimedia (developing student skills)	2.45	2.67
distance learning	1.45	2.20
classroom teleconferencing	1.04	2.04
student laptop ownership	1.24	2.42
virtual classroom	1.06	2.22

Figure 4. Top Five Motivators and Impediments for Each Technology

	Ranking by Each Technology								
Top Motivators	E-mail	Wired	Multi-media	Distance learning	Class-room teleconf.	Laptop ownership	Virtual library	Virtual classroom	Count*
Competitive adv. with peers		4	3	4	2	2	4	2	7
Leading edge of technology		1			1	1	1	1	5
Appeal to new students			5	3	4	4		3	5
Access to data on campus	3	2				3	2		4
Part of school's mission	1		1	2					3
Comm. with people off-campus	1			5	3				3
Access to data off-campus	2	3					3		3
Increased teaching productivity	4			1				4	3
Class time productivity						5		5	2
Ability to present concepts		5	2						2
Ability to gain insights			4		5				2
Faculty demand							5		1

	Ranking by Each Technology								
Top Impediments	E-mail	Wired	Multi-media	Distance learning	Class-room teleconf.	Laptop ownership	Virtual library	Virtual classroom	Count*
Funding/money	2	1	1	1	1	1	1	1	8
Perceived benefits	5	3	4	2	2	2	4	5	8
Faculty interest	1	5	2	5	3	3		2	7
Commit. from school admin		4		3	4	4	2	3	6
Faculty training	3		3				3		3
Infrastructure		2			5	5			3
Easier access from remote	4								1
Equip to use on classroom			5						1
Part of school's mission				4					1
Commit. from campus admin								4	1
Tech. less complicated							5		1

* Count = number of times (out of eight technologies listed in headings) item made top five list (out of twenty-two motivators or impediments)

Providing a competitive advantage with peer institutions was listed as one of the top five motivators for all of the technologies except e-mail.

Clearly pedagogical motivators (ability to present concepts not possible otherwise, ability to gain insights not possible otherwise, increased class time productivity) rarely made the list of top five motivators. Faculty demand only made the top five list for virtual libraries, the one technology where faculty interest was not an impediment. Reduced overall cost of instruction over time never made the top five list of motivators for using any of the technologies. It was number six for distance learning

Unfortunately, there was no statistically significant difference between tiers for motivators or impediments for using any of the technologies. In other words, there do not seem to be any strategies used by higher tier schools that could be passed on to others for using IT more effectively. One other finding from this study is that there was no relationship between the teaching versus research orientation of institutions and their use of technology either. The only technology that did have such a relationship was e-mail, and it was used more by more research-oriented schools. This suggests that e-mail is viewed more as a tool for faculty to do research than to improve teaching.

4. Colleges and universities are generally not focusing on pedagogical or productivity-related reasons for adopting information technologies. Multimedia technology was the only one where clearly pedagogical reasons made the top five list. Reducing the overall cost of instruction over time never made the top five list of motivators. The top two motivators cited - having a commitment to staying on the leading edge of instructional use of technology and providing a competitive advantage with peer institutions - are clearly based on having the technology itself versus improving the teaching and learning process or decreasing costs.

These results contrast sharply with reasons industry invests in IT. The following quote is taken from a popular textbook used in many college courses on the use of information systems in organizations.

> When new information systems technology was first introduced, managers tended to adopt the technology first and then tried to figure out what to do with the new information and cope with the business and organizational implications. ***Such an approach is now recognized as woefully inadequate***...Companies successful at adopting new technology recognize that the key to capturing real business benefits is the role that business managers take in leading the introduction of information systems. (Reynolds, 1992).

Reynolds also mentions the two key approaches companies now use to justify investments in IT - support of key business strategies and net present value analysis (or similar financial measures). A major study done by James Bacon in 1992 supports Reynold's research. Bacon received survey responses from executives in eighty different firms. Figure 5 summarizes the key reasons (motivators) respondents reported for investing in new information technologies. Note that business/financial reasons had much higher rankings than technical ones. Introducing new technology received the lowest ranking by leaders in industry. This contrasts sharply with the top motivators for using technologies listed by leaders in higher education as shown in Figure 4.

Figure 5. Ranking of IT Investment Criteria for 80 Firms (Bacon, 1992)

Criteria/Reason for Investing in IT Projects	Ranking Based on Overall Value of Projects
Supports explicit business objectives	1
Has good Internal Rate of Return (IRR)	2
Supports implicit business objectives	3
Has good Net Present Value (NPV)	4
Has reasonable payback period	5
Used in response to competitive systems	6
Supports management decision making	7
Meets budgetary constraints	8
High probability of achieving benefits	9
Good accounting rate of return	10
High probability of completing project	11
Meets technical/system requirements	12
Supports legal/government requirement	13
Good profitability index	14
Introduces new technology	15

In contrast to the business-driven focus for using IT in industry today, the majority of college campuses seem to be using the *woefully inadequate* approach described by Reynolds by focusing on the technology first. This suggests that colleges are not deciding on specific needs first and then adopting the appropriate technology; they are adopting new technologies and then trying to figure out what to do with them. It also shows that people inside of higher education have very different intentions for using IT than people outside of higher education.

Conclusions

In general, the conditions which many organizations (private firms, in particular) have found to support positive returns for investing in information technology include taking a more business-driven or strategic approach to using technologies, providing strong leadership and planning for IT, using IT to reengineer key processes, and ultimately incorporating the technology within the organization's operational and/or managerial work systems (Bacon, 1992, Champy and Hammer, 1992, Reynolds, 1992, Wiseman, 1985, Zmud and Apple, 1992).

Institutions of higher education seem to be repeating many of the mistakes made by industry in this area. In particular, colleges and universities in this sample of 108 national institutions appear to be adopting technologies first and then trying to figure out what to do with them and how to cope with the organizational implications. Colleges and universities do not appear to be using IT to help reengineer key processes (in particular the teaching and learning process), and a key problem appears to be the conflicting visions of the purpose of most institutions (research versus teaching) as well as the reward structures (for research versus teaching) and therefore the role IT should play. People outside of higher education are also much more concerned with costs than people inside higher education.

There is very little formal planning for IT, as evidenced by the lack of commitment from school administration being cited as a major impediment for many of the technologies. It also appears that institutions would purchase more of all technologies if they had more money, even though they question their perceived long term benefits and educational value. It clearly seems that basic MIS principles could benefit higher education's use of IT.

References

Astin, Alexander. What Matters in College? Four Critical Years Revisited. San Francisco: Jossey-Bass, 1992.

Bacon, James C. "The Use of Decision Criteria in Selecting Information Systems/Technology Investments," MIS Quarterly, Volume 16, Number 3, Sep 92.

Frand, Jason, and NG, H. Alvin. "Eleventh Annual UCLA Business School Survey: Use of Learning Technologies in Business Education," Regents of the University of California, June, 1994.

Geoghegan, William H. "What Ever Happened to Instructional Technology?" Norwald, CT: IBM Corporation, 1994.

Green K. C. and Eastman, S. (1994). "Campus Computing 1993: The USC National Survey of Desktop Computing in Higher Education," University of Southern California, Los Angeles, CA.

Hammer, Michael & Champy, James. Reengineering the Corporation, New York: Harper Business, 1993.

Heterick, Robert C., Jr., ed. "Reengineering Teaching and Learning in Higher Education: Sheltered Groves, Camelot, Windmills, and Malls," Professional Paper Series, no. 10, CAUSE, Boulder, CO, 1993.

Krueger, Alan B. "How Computers Have Changed the Wage Structure: Evidence From Microdata, 1984 - 1989," *The Quarterly Journal of Economics*, February, 1993.

Reynolds, George W. Information Systems for Managers, West Publishing Company, St. Paul, MN 1992.

Stoll, Clifford, Silicon SnakeOil, Second Thoughts on the Information Highway, New York: Doubleday, 1995.

Twig, Carol. "Can Education Be 'Productive'?" *EDUCOM Review*, Vol. 28, no. 6, November/December, 1993.

U.S. News & World Report, America's Best Colleges, 1995 College Guide, Washington, D.C.

Willut, C. Posting to AAHE Technical Activities and Projects Internet list, March 8, 1994.

Wiseman, Charles. Strategy and Computers: Information Systems as Competitive Weapons. Homewood, IL: Dow-Jones-Irwin, 1985.

The Wingspread Group, "An American Imperative: Higher Expectations for Higher Education," The Johnson Foundation, 1993.

Zmud, Robert W., and L. Eugene Apple. "Measuring Technology Incorporation/Infusion," *Journal of Product Innovation Management*, Vol. 9, No. 2, June, 1992.

PANEL SESSION
ETHICAL ISSUES IN THE SOFTWARE DEVELOPMENT WORKPLACE

MODERATOR AND PANELISTS:
Joyce Currie Little, Professor
Department of Computer & Information Sciences
Towson State University
Baltimore, MD 21204
(410) 830-3783 fax: (410) 830-3868
jclittle@midget.towson.edu

Ronald Kizior, Professor
Management Science Dept.
Loyola University Chicago
820 N. Michigan Ave.
Chicago, IL 60611
(312) 915-7394
rkizior@wpo.it.luc.edu

Netiva Caftor, Professor
Northeastern Illinois University
Dept. of Computer Science
5500 N. St. Louis
Chicago, IL 60625
(312) 794-2739
uncaftor@uxa.ecn.bgu.edu

There are many ethical issues that may arise within the software engineering workplace. Some of them are not unique to the technical community, being of a more generic nature. Others, however, are unique to the practice of software development and may have serious impact on the product, the users, or society. This panel will address some of the ethical issues, both technical and generic, that may arise in the software development environment in the workplace.

The moderator will describe some of the work of the Software Engineering Ethics and Professional Practices (SEEPP) Task Force, formed by the IEEE-Computer Society and ACM by the Committee for the Establishment of Software Engineering as a Profession. Examples of ethical dilemmas that may face the entry-level graduate upon becoming a practicing software engineer will be given, along with a suggested classification of such problems. The question of whether the software development industry could be improved by having enhanced monitoring or controls, such as certification and/or licensing of personnel or products, will be discussed.

There is a continuing interest in including instruction about ethical issues in courses on software development and in the curriculum overall. The second panelist will tell about experiences in teaching ethical issues in classes in a COBOL software development environment within a School of Business. Exercises based on the "Killer Robot" materials will be described and evaluated.

Gender issues concerning equity and access to computing have arisen in many locales, and the decline in the number of women entering computer science as a major has concerned many. The third panelist will discuss gender issues that may arise within a software development team, based on experiences with gender research, and will relate these to some of the ethical issues being discussed. Research questions about the role of gender in the software development team, the process and the workplace will be addressed, such as "How well do women and men work together in a team environment?", "Are women more aware of ethical issues than men?", and "Are some aspects of the software development work more natural for women than for men?"

What Constitutes "Exceptional" in Information Systems Teaching?

David L. Russell, Associate Professor and Chair
Anil Gulati, Associate Professor and Assistant Dean
Quantitative Methods and Computer Information Systems Department
School of Business
Western New England College
Springfield, MA 01119-2684

drussell@wnec.edu, agulati@wnec.edu

ABSTRACT

Increasingly, college teachers are called upon to document their effectiveness. Yet, there is little qualitative and virtually no quantitative guidance support what constituted excellence in college teaching. This problem is compounded when it comes to identifying exceptional IS teachers. Exceptional IS teachers possess many of the characteristics of exceptional teachers generally. Information Systems teachers face an additional challenge, it is argued here, in that there are a series of expectations they must satisfy that differentiate the exceptional IS teacher from IS teachers in general. Six such areas are identified and discussed.

INTRODUCTION

Accountability and assessment are bywords in education today. Increasingly, college faculty members must demonstrate that they do their jobs well *and* be prepared to document that they do there jobs well. This is true all the more at "teaching" institutions, which place a premium on quality teaching.

The authors' institution is one such college. For many years, faculty members were evaluated on their teaching. The percentage of their total evaluation allotted to teaching could range from 50 to 65%, with Advising, Governance and Professional Development/Scholarship constituting the remainder of the evaluation model. Teaching could be evaluated as "Meritorious", "Acceptable" or "Unacceptable" (Western New England College, 1986). In turn, the meritorious rating was defined as "good to great".

Over several years, the *Faculty Handbook* was revised (Western New England College, 1995). In the area of faculty evaluation, several changes took place. The number of categories was expanded to five by separating the Professional Development and Scholarship/Research into separate categories. The percentage to be allocated to Teaching became 55% to 65% and an extensive definition of "Teaching Effectiveness" was developed, constituting more than a page of the Faculty Handbook. Most significantly, a new category for evaluating teaching was added: "Exceptional". Even more significantly, to quote the *Faculty Handbook*, "to be rated as 'exceptional' in teaching, faculty members must be clearly and significantly superior to most of their colleagues... Individual departments will devise criteria for exceptional teaching."

Thus, the Information Systems ("IS") Department, in the Fall of 1995, tasked itself with the definition of an exceptional IS teacher. This proved to be more challenging than first envisioned. This is because an exceptional IS teacher is, first and foremost, an exceptional teacher, and thus shares certain characteristics with all exceptional teachers.

EXCEPTIONAL TEACHING

There is a rich literature concerning quality in teaching, and a significant portion of it focuses on teaching at the college level. For the most part (and in contrast to the literature on good teaching at the elementary and secondary levels), this literature is anecdotal in nature. Milton (1978), for example, consists of fourteen chapters written by well-known teachers and discussing various aspects of the teaching profession. Ericksen (1984), in contrast, is a small volume setting forth this one individual's definition of good teaching. As valuable as these are (particularly Ericksen's first chapter), they do not provide a concise and quantifiable measure of excellence in teaching.

Morrill and Spees (1982) take a more descriptive approach. In describing an excellent teacher, they include such factors as:

- To recognize the teaching process as an art.
- To be able to adjust to individual differences among students.
- To encourage and maintain the native desire for knowledge.
- To work to increase the student's ability to solve problems.
- To communicate comfortably and easily with individuals and groups.

(Morrill and Spees, p. 83-84). In doing so, they develop a comprehensive model of good teaching and go on to discuss a wide variety of related professional issues. Although dated, Hamacheck provides perhaps the most comprehensive exposition of the characteristics of good teaching (Hamacheck, 1969).

More recently, Smith and Simpson used the Delphi Method to determine twenty-seven competencies viewed as important to a panel of academic leaders nationwide. The top ten, in rank order, are as follows:

1. Provide helpful feedback
2. Exhibit respect and understanding
3. Demonstrate mastery of subject
4. Communicate effectively in writing and orally
5. Develop a "reflective approach" to teaching through collecting feedback and continually modifying instructional approaches
6. Promote individual involvement of students
7. Enhance motivation through personal enthusiasm for the subject
8. Communicate and manage appropriate expectations
9. Demonstrate a general belief that all students are capable of learning
10. Encourage cooperation and collaboration among students.

(Smith and Simpson, 1995). By extension, a high caliber of performance on all or most of these dimensions would constitute a definition of excellence in teaching. However, it must be recognized that most college instructor's experience is limited with respect to training in teaching or in teaching methodologies, and thus are likely not to be exposed to this literature.

EXCEPTIONAL "IS" TEACHING

An exceptional IS teacher is, first and foremost, an exceptional teacher. As such, she or he possessed a large number of the characteristics suggested in the above scan of the literature.

It quickly became apparent, however, that an exceptional IS teacher possesses certain characteristics, behaviors and qualities that go beyond excellence in teaching generally. The purpose of this paper is to address those factors, and thus to answer, at least partially, the title's question: what constitutes "exceptional" in *Information Systems* teaching? In doing so, we hope to generate a discussion among our colleagues that will be enlightening to ourselves, and particularly to those administrators who evaluate IS teachers. For discussion purposes, we have divided these characteristics into six areas:

1. Currency in the field;
2. Degree of student support required;
3. Need to motivate students;
4. The sequential nature of the IS curriculum;
5. Adaptability in the face of imperfect technology;
6. Serving as a technology resource person.

Each of these factors will now be discussed separately.

1. CURRENCY IN THE FIELD:

All teachers and scholars must keep current with their field, and a teacher who is "out of date" is unacceptable. All fields of knowledge have experienced a blossoming of new research and knowledge in the latter half of the century. Few fields, however, have had the exponential explosion of knowledge that Information Systems and its parallel fields of Computer Science and Electronic Engineering has experienced. This statement is equally true for any meaningful subdivision of the field.

This is further compounded by two factors. The first factor is the rise of completely new subdivisions in the field regularly, and their relatively quick transition into commercial use. In recent years, for example, we have seen the rise of End User Computing, the Client-Server Model and relational database systems, all of which did not exist commercially, and barely existed theoretically, a generation before. The case of relational database can be illustrative. The seminal paper appeared in 1970 (Codd, 1970). By 1979, the ANSI/X3/SPARC DBS-SG Relational Database Task Group had been chartered to define an industry standard for the new database model. By 1982, no fewer than fourteen database systems could be considered relational, at least partially

(Schmidt & Brodie, 1983). And, by the late 1980's, the relational database model was clearly established as the industry's "leading edge" and was the dominant model used in college teaching. Thus, less than twenty years separated the seminal paper and the widespread application of a technology, which the college instructor is expected to master and teach properly. It would not be difficult to find examples of technologies achieving dominance in even shorter time periods. Few academic areas must change and adjust so quickly.

The second factor is the fact that "old" technologies and system do not fade away, but are preserved and extended. Thus, the IS teacher must teach students to be competent in technologies that are nearly forty years old (e.g., transaction processing in COBOL) and in technologies that are very new (e.g., Windows 95). The example in the preceding paragraph can be extended here. Despite the fact that the relational database is now the established standard for college teaching, the fact remains that older database models still do a great deal of work in industry. Thus, the well-prepared IS student must be exposed the older technologies of hierarchical systems and the CODASYL or network model. If that student went to work in a high-volume, transaction-oriented environment (e.g., financial services or the insurance industry), she or he would likely work in one of the older technologies because of its executional efficiency. These "legacy systems" abound.

The actual volume of new knowledge to be acquired, digested and presented to students is itself a major challenge. Perhaps even more challenging is the series of paradigm shifts the field has experienced (Kuhn, 1982). We have moved from centralized "big iron" systems to highly dispersed PCs and back to shared networks within half a generation. Sometimes, the paradigms fundamentally conflict, as when "record at a time" paradigm of third generation programming languages must integrate with the table instantiation paradigm of relational databases. IS teachers, being human, must go through these same paradigm shifts, and more significantly must help our students deal with multiple paradigms simultaneously.

Thus, a person who can keep abreast of new technologies and changing paradigms and who can incorporate the revised understanding into his/her courses is truly an exceptional teacher. Most of us have adopted a coping strategy of focusing on one segment of the field, and even then keeping current is a great challenge. Few IS Departments, however, are so large as to permit extensive specialization. Most IS instructors must teach a wide variety of courses. Keeping current remains a staggering challenge.

There is yet another aspect of currency which challenges all IS instructors, and at which the exceptional IS instructor excels. This is the ability of learn new technologies easily and repeatedly and then to incorporate those technologies into one's teaching very soon after acquiring it. In other fields, a new technology will be announced, discussed and adopted in the commercial world over many years. Then, colleges will react by incorporating the material into their curriculum.

Not so in IS. First, the cycle time between the introduction of technology, its commercial acceptance and the need to incorporate it into the curriculum is significantly shortened. Curriculums that feature technologies that are even a few years old are decidedly old-fashioned. Second, even well-established technology can undergo rapid, even radical change. This is most frequently seen in software, where a new, upgraded version is so different from the preceding version that is renders textbooks and the teacher's knowledge obsolete.

The exceptional IS teacher accepts this situation and had developed learning skills that permit him/her to rapidly learning new technology and to do this repeatedly. Further, his/her learning skills allow the teacher to very rapidly teach the material. In fact, these two processes often occur nearly simultaneously, with the learning process only slightly preceding the teaching process. Few other fields make this demand upon instructors.

2. DEGREE OF STUDENT SUPPORT REQUIRED:
All good teachers go out of their way to help students; the exceptional teacher can elicit from students their needs for help and respond to them. The IS teacher faces an additional challenge. Unlike other fields, learning in the IS field is neither intuitive nor facile. Students in other subject areas, such as writing and mathematics, often receive support services in such as Writing Labs and mathematics tutoring. These support services are often nonexistent in the IS curriculum, indicating that the IS instructor must provide greater-than-average support service. The shifting and, at times, conflicting paradigms of the field frequently snare the unwary student.

The IS teacher faces a particularly challenging situation in that IS students face a *dual* learning curve. First, the IS student must master the intellectual material of the course. In this, IS students face the same burden as in any other course. Second, however, students must master the technology employed in the course. These are separate and distinct learning curves, and this reality is poorly understood by those not acquainted with our field.

For example, a COBOL student must simultaneously master two distinct concepts: first, s/he must develop the intellectual concepts of sequence, decision and iteration as the core blocks of computer programming. Simultaneously, the student must master the nuances and quirks not only of COBOL itself, but of the specific compiler implementation present at his/her college. In other words, the student must comprehend both the *semantics* of the subject matter as well as the *syntax* of the language used to implement the subject matter. It is as if a history student were simultaneously learning history and learning to read.

A further example can be found in Desktop Publishing. Here students have to learn the intellectual concepts of design, layout, typography, and the like. Simultaneously, they must learn the technology of the desktop publishing software, laser printing, scanning and the like. And, since most desktop publishing packages are Windows-based or Macintosh-based, many students face a third learning curve, as desktop publishing is often their first exposure to Windows, File Manager, Print Manager, and the like (or their Macintosh equivalents). In few other fields do instructors teach, and students learn, on multiple learning curves simultaneously. The opportunities for students to become confused are abundant, and few students pass up the opportunity.

Thus, we submit, the exceptional IS teacher will spend more time directly supporting students than teachers in other subjects, and will spend that time most effectively, though not most productively, in one-on-one work with the student.

3. NEED TO MOTIVATE STUDENTS:

The IS field is exciting, and job opportunities are plentiful, but students in recent years have not been fully aware of this. We can speculate on many reasons for this, including the fact that commercial IS software is not as enticing as the entertainment software with which students have grown up. More significantly, however, is the fact that IS material does not come easily: the learning curve can be steep and protracted.

This is so for two key reasons. First, as discussed above, mastering IS material typically involves two learning curves simultaneously: the content itself and the technological implementation of the content. Second, computers simply do things differently than humans do. The massive effort to develop metaphors that will suit end users, such as the spreadsheet metaphor or the table metaphor used in relational database, is evidence of the fact. IS students, however, must go beyond the relatively facile metaphors seen by end users and into the underlying structures, which are not at all facile.

For example, a user interface may have a user instruct the system to "find all sales reps with sales greater than average". The IS student, however, knows that at the programming level, this simple act involves :

1. finding the correct file;
2. setting up two variables, one to accumulate total sales and one as an incremental accumulator (a "counter");
3. sequentially scanning through the underlying file to accumulate the total sales and the total count;
4. computing the average by dividing the latter into the former;
5. reset the search to the top of the file;
6. sequentially scanning through the file once more, comparing sales of each representative to the newly-computed average sales;
7. for each representative, determine whether the sale is greater than average, storing those who are above average to a temporary table; and
8. transferring the temporary table to the output devise in a format acceptable to the user.

Thus, what is intuitive or facile to the end-user is anything but intuitive or facile at the level at which the IS student works. At the same time, however, the student must keep in mind the user's perspective as well as their own, technical perspective. This dichotomy of perspectives constitutes much of what makes being an exceptional IS teacher challenging.

Thus, the exceptional IS teacher is one who can motivate a student about the opportunities of the field, keep the student motivated while on their several learning curve and sustain the student over the inevitable glitches that occur in technological environments.

4. SEQUENTIAL NATURE OF THE IS CURRICULUM:

Many subjects build up course matter cumulatively, but IS does this to an extraordinary degree. This is because the more fundamental material in IS courses involves technological steps that cannot be worked around or compensated for. For example, a COBOL student must write the Data Division accurately and precisely before the Procedure Division can be expected to work properly. Moreover, national standards (e.g., Data Processing Management Association, 1991) mandate a highly sequential and cumulative approach to the material from course to course. As a result, IS faculty members in later courses have very specific expectations of student attainment in earlier courses. This places the IS instructor teaching the earlier course under great pressure to deliver the student to the needed point of development, and consequently limits his/her freedom to teach what and how s/he wants. Thus, IS instructor does not have luxury of a variable course ending point, but must stay on syllabus schedule and cover what is expected of him/her.

In addition, knowledge of IS material is more absolute than in many other fields: either the student can perform or s/he cannot. Moreover, we teach material that will be evaluated by our students' employers. Those employers have a right to expect that we have covered certain material. Since the skills and knowledge we teach is valued by employers, we cannot skip or slight any steps. Thus, we face a great deal of pressure to accomplish what is stated in our syllabi (often derived from national standards). The result is that the exceptional IS teacher must explain each concept well, structure each component into a cohesive whole and knit the course together in such a way that students will understand each steps role and its relationship to the whole. Simultaneously, s/he must cover all of the syllabus and keep students motivated for the entire course. In few other fields do instructors face such a comprehensive challenge.

A key way of motivating students is to allow them to demonstrate what they know. Because of the sequential nature of the IS curriculum, this is particularly challenging for the IS teacher. The technology often mandates that an significant amount of work be done prior to having any demonstrable result. For example, the COBOL instructor must define, describe and teach file structures and independent variable concepts prior to creating a Data Division, which in turn must proceed the writing of the Procedure Division, which in turn must be compiled. This is compounded by the fact that COBOL is a particularly difficult language from a conceptual, syntactical and semantic point of view. Further, each COBOL compiler has its own methods, editors, and quirks.

The exceptional IS teacher nonetheless devises methods for students to demonstrate their knowledge as the course progresses. This is particularly important to motivate students, as discussed in the preceding section. By demonstrating that they are truly learning, students move more quickly along the learning curve.

5. ABILITY TO DEAL WITH IMPERFECT TECHNOLOGY:

Few fields are as dependent on technology for teaching as IS. The IS instructor will frequently use technology in class to teach, demonstrate or experiment with the subject matter. Equally frequently, students will use the technology in class as part of the learning process and, not infrequently, as part of the evaluation process. Such a situation, however, is vulnerable to technology failure.

The exceptional IS teacher has developed coping strategies to deal with the inevitable failures. These strategies center on two complimentary areas. First, the instructor can become expert with the teaching technology, allowing the instructor to debug problems "on the fly". Second, the instructor can always have a second, alternative class on hand to switch to when the need arises. Few other fields expect their faculty to become knowledgeable about the teaching technology, as opposed to their subject matter, and few other fields expect their faculty to, in effect, prepared two lectures for every class period. This reality is poorly understood by those outside our field. In particular, those who evaluate IS teachers must comprehend this portion of our reality and take it into account when evaluating IS teachers.

6. SERVING AS A TECHNOLOGICAL RESOURCE PERSON:

Of course, IS faculty are not alone in their use of computing technology in the curriculum. Information Systems have diffused throughout the curriculum and are used increasingly by faculty in many different fields of study. These faculty need support, often beyond the capabilities of the college's in-house support staff. As a result, these faculty look to their IS colleagues as in-house resource persons. Thus, IS faculty often serve a technical resources for our colleagues, our administrators and, often, students not in our classes.

Being asked for advice in one's areas of expertise by a colleague is not an unusual experience for many faculty members. What differentiates an IS faculty member is both the volume and the constancy of support requests from colleagues and administrators. The exceptional IS faculty member accepts this situation, helps to the degree needed and, virtually always, does it willingly and without charging the usual consulting fees! Further, those who evaluate IS instructors tend to have little understanding of the depth, degree or constancy with which IS instructors are called upon to support their colleagues technically. Yet, if the help from IS is not forthcoming, highly negative consequences will occur. These consequences begin with upset colleagues, but do not end there. Rather the IS instructor is motivated to encourage the use of IS technology elsewhere in the curriculum not only to improve the curriculum, but to build a body of support for the very technology which the IS instructor needs to do his/own own job. Thus, exceptional IS instructors can be expected to give an extraordinary amount of help and support to colleagues. What is necessary is to educate academic administrators on this point.

CONCLUSION

An exceptional IS teacher is first and foremost an excellent teacher. This paper has addressed six area which differentiate an exception IS teacher from exceptional teachers in general.

To further discussion on this topic, we have presented six distinct areas in which an exceptional IS teacher must excel in addition to being an exceptional teacher in the first place. They are: the need for currency in the field; the degree of student support required; the need to motivate students; the extraordinarily sequential nature of the IS curriculum and its implications; the need for adaptability in the face of imperfect technology; and the need to serve as a technology resource person.

This list is not intended to be complete or exhaustive. Rather, its sole purpose is to stimulate thought and discussion among IS teachers. A result will be a marshaling of our arguments in order to educate those who evaluate our teaching as to the unique demands placed upon the exceptional IS teacher.

REFERENCES

Codd, E.F., "A Relational Model of Data for Large Shared Data Banks", *Communications of the* ACM, 13:6 (June, 1970): 377-387

Data Processing Management Association, *Information Systems: the DPMA Model Curriculum for a Four Year Undergraduate Degree,* DPMA, 1991

Ericksen, Stanford C., *The Essence of Good Teaching: Helping Students Learn and Remember What They Learn*, Jossey-Bass, 1984

Hamachek, D. "Characteristics of good teachers and implications for teacher education", *Phi Delta Kappan*, 1:6 (1969): 341-345

Kuhn, Thomas S., *The Structure of Scientific Revolutions,* University of Chicago Press, 1982

Milton, Ohmer, *On College Teaching: A Guide to Contemporary Practices*, Jossey-Bass, 1978

Morrill, Paul H. and Emil R. Spees, *The Academic Profession: Teaching in Higher Education*, Human Sciences Press, 1982

Schmidt, Joachim W. And Michael L. Brodie, *Relational Database Systems: Analysis and Comparison*, Springer-Verlag, 1983

Smith, Kathleen S., and Ronald D. Simpson, *Innovative Higher Education*, Human Sciences Press, 1995

Western New England College, *Faculty Handbook*, Eight edition, 1986, and Ninth Edition, 1995

ACKNOWLEDGEMENT

The work presented here derived from the hard work of the author's colleagues in the QM/CIS Department at Western New England College, as the original ideas stemmed from a group discussion. The authors' colleagues, in alphabetical order, are Profs. Jerzy Letkowski, Clara Lim, C.Y. Lim, Marilyn Pelosi and Instructor Casimir Storozuk. The authors acknowledge their contribution to this paper with gratitude.

Recruiting Computer and Information Sciences Students: Lessons Learned

Sharon N. Vest and David L. Feinstein
University of South Alabama
Mobile, Alabama
334-460-6390
vest@cis.usouthal.edu

ABSTRACT

Technical disciplines, such as computing sciences and engineering, have struggled over the years with the recruiting and retention of students. Interest in the computing sciences grew steadily through the 1960s and 1970s. In the early 1980s, there was a phenomenal period of growth. This may be attributed to the success of the space program and the introduction of the personal computer. Just as quickly as they came, they left. Many factors have contributed to the decline, ranging from a lower college-age population, to poor quality of preparation for advanced education in the sciences and technical disciplines. This paper is based on a collection of techniques that have been attempted and, in most cases, successfully used. We believe that some variation of these ideas can be incorporated into recruiting and retention strategies in other academic units. Dynamic strategies are required, especially as we enter the "next generation"--multimedia, homepages and the Web!

Background

Technical disciplines, such as computing sciences and engineering, have struggled over the years with the recruiting and retention of students. Interest in the computing sciences grew steadily through the 1960s and 1970s. In the early 1980s, there was a phenomenal period of growth. This may be attributed to the success of the space program and the introduction of the personal computer. Just as quickly as they came, they left. Since about 1983, there has been a slow decline in interest. Many factors have contributed to the decline, ranging from a lower college-age population, to poor quality of preparation for advanced education in the sciences and technical disciplines.

At a time when fewer students are choosing technical careers, demand for technically trained employees is on the rise. Nationally, companies are downsizing and, as a result, entry level skill demands of prospective employees have changed. Employers are recruiting employees with technical skills. Research indicates that graduates of master's degree programs possessing technical skills are significantly more marketable and receive higher salaries than their non-technically trained counterparts.[Obeman, Nagle]. On every predictive list of careers for the future, computer related positions rank at the top. The March 1995 issue of Money magazine listed the 50 fastest growing careers of the future, four of which were in computer science. Computer engineer and systems analyst ranked one and two. U.S. News and World Report [October 1994] placed operations research analyst, computer engineer and information services in its list of the "Top Ten Hottest Job Tracks" for the future. Recruiting of students into computer science programs is not only vital to the discipline, but imperative to meet the demands of the national job market and economy. Recently, renewed interest in the computing sciences has emerged--the popularity of Internet being the speculative cause.

Within our University, the School of Computer and Information Sciences (CIS) is an independent academic unit. It is not affiliated with many of the more natural homes in which computing programs are found, such as engineering and business. Consequently, recruiting and retention have always presented unique and interesting challenges. This paper is based on a collection of techniques that have been attempted and, in most cases, successfully used. We believe that some variation of these ideas can be incorporated into recruiting and retention strategies in other academic units. Dynamic strategies are required, especially as we enter the "next generation"--multimedia, homepages and the Web!

Recruiting has been a top priority since 1987 when enrollments dropped dramatically. In 1985, the School had 635 majors, by 1987 that number had decreased to 490. A committee was appointed to tackle the problem with full support from the administration. The committee, through trial and error, developed a recruiting strategy incorporating a variety of activities. Success has been demonstrated by a stable enrollment over the years since 1987, in spite of a decreasing pool of high school graduates and college age population (ages 18-24) nationally. [National Center of Education Statistics] [Figure 1]

The committee worked closely with the University's Department of Admissions. Two goals emerged from the sessions: (1) recruit new students, first time freshmen and transfer students; and (2) retain students beyond the first year curriculum.

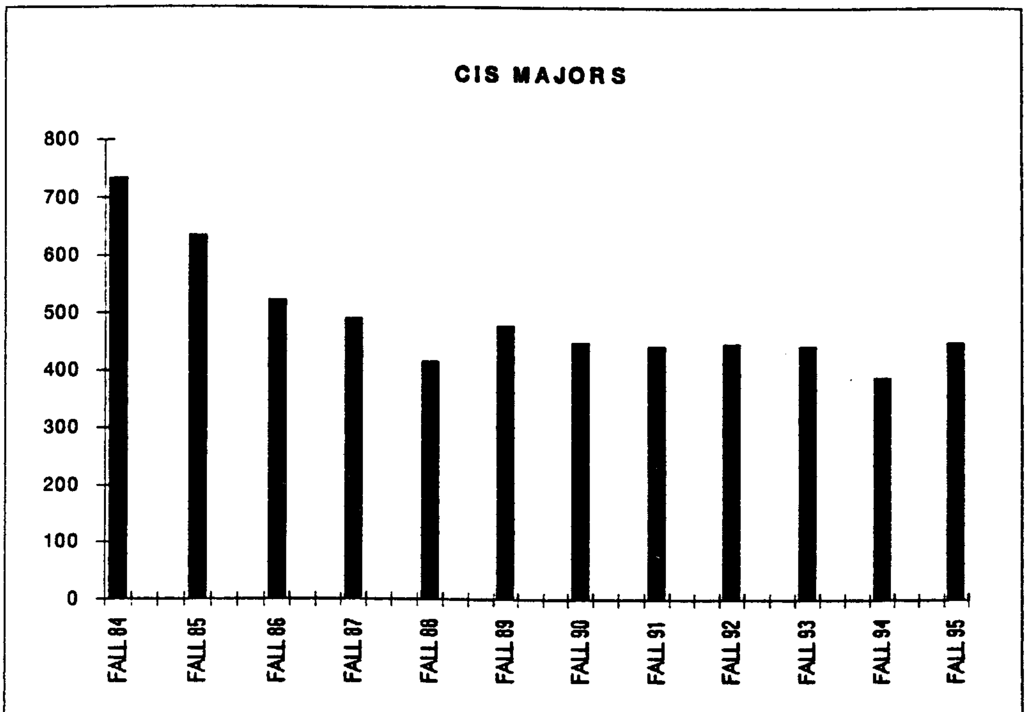


Figure 1

Recruiting

The committee concluded that "accessibility" was vital to successful recruiting. The School began a campaign to advertise its accessibility--the willingness to share our expertise and resources in the area of computing sciences. Whenever feasible, service "on demand" was immediately provided. If someone wanted a tour of the computer facilities, we arranged it, and served refreshments. If a group wanted a workshop on WordPerfect (or almost anything else), we arranged it *and* served refreshments! Faculty representatives attended local and regional educational conferences to become "visible" and accessible to teachers and counselors of potential students.

Several specific recruiting strategies were developed:

(1) CIS Day
(2) Speaker Service
(3) Community College Articulation Agreements
(4) Workshops on various topics: Internet, Word-processing...
(5) Active Participation in Professional Activities
(6) Collaboration with the University's Admissions Department

During the first few years of recruiting experiments, the School conducted "CIS Days." Our objective was to invite potential students from the area to our campus and "excite" them about careers in computer and information sciences. Invitational letters were sent to all local high school computer science teachers inviting their classes to a day of organized CIS activities on campus. The students were grouped, and each group rotated among activities planned and staffed by the faculty. These activities included: (1) Exploring the Internet; (2) Introduction to Spreadsheets; (3) Introduction to Word-processing; (4) Campus Tours; (5) CIS Curriculum and Expectations and, (6) Advanced Student Project Demonstrations.

The feedback from the CIS days was encouraging. The students appeared interested and their teachers appreciative, but the process was not without problems:

1. We were inundated with requests to bring multiple classes from each school.
2. Screening for appropriate population was difficult.
3. The cost in terms of time and resources was extremely high.

The response to the letters was very positive—in fact, overwhelming. Most teachers wanted to bring their entire class. Sixty was the maximum number of students that could be handled comfortably; therefore, it was very difficult to accommodate all requests--not good for public relations! Although the letter specified that teachers select students with both interest and aptitude for the discipline, often the entire class would attend, many of whom appeared more interested in the pizza lunch that the rotation of activities! For the next few years, we restricted the invitational letters to advanced math and science instructors. This effort brought a more homogeneously appropriate group to campus. It was still very difficult to accommodate all the requests. The price in terms of faculty time and effort per student was very high. Interviews with those in attendance showed few potential candidates for our program. CIS Days are now conducted on an occasional basis, either as a result of a special request, or to target a much smaller, select group of students.

In 1988, the School initiated a speaker program for local high schools. Letters listing specialization areas of the faculty were sent to all local high schools volunteering to speak on any of a number of topics. Special topic requests were also honored, whenever possible. Over the years, this service evolved into a "Careers in the Sciences" presentation given regularly to local high schools by various faculty members. The presentation consists of a set of components from which to customize for a particular request. The components include a video on a current or popular topic, such as *Virtual Reality*, and a PowerPoint presentation covering job market predictions for the next decade, basic degree requirements, and the University's strong points. A contest quiz is also given with prizes for students with the highest scores. The quiz calls for students to answer questions about careers and to design (in groups) a solution to some classic CIS problem, like the "Cannibal and the Missionary Problem" or the "Basket of Eggs Problem." This approach has been extremely successful. Faculty members can rotate conducting the presentation, choosing appropriate components to fit the time frame and target population.

To recruit transfer students from the community colleges in Alabama, as well as those in neighboring states, articulation agreements were developed. Personal visits to feeder colleges were made over a 3-year period by the School's chairperson and the committee chairperson. Formalized articulation agreements were developed with each college after discussing curriculum issues and course content with their faculty. The community colleges were sincerely appreciative of our concern for the success of their students. The School used these opportunities to advertise our program, discuss future plans and establish communication channels between the feeder institutions and the University.

The School continues to sponsor customized workshops on campus to increase our visibility in the community. The goal of these workshops is to motivate the educator participants to be messengers of the advantages of technical education, and reveal how well equipped our institution is to provide that education. These workshops include such topics as: (1) Using Internet as a Research Tool; and (2) Using Microsoft Word. Audiences include various groups from the local public and private school systems and professional organizations. Again, the emphasis is on accessibility.

The School faculty frequently participates in University sponsored recruiting functions. Faculty are present to answer questions from parents and students. These functions include high school receptions throughout the state, and educator conferences held on campus. Specialized CIS brochures are designed annually, and included in Admissions mailings to prospective students. Educating potential students about the opportunities in computing sciences continues to be a major objective.

Assuming a leadership role in educational organizations, Alabama Council for Computers in Education (ACCE) and the Association of Computing

Machinery (ACM), was also seen as a recruiting strategy. The high visibility helped create a constructive rapport with high school and community college faculty in the region. Our School has played a leadership role in curriculum development. Faculty efforts have resulted in major National Science Foundation grants for curriculum development and faculty training. Additionally, our faculty have become major players in Information Systems curriculum development. These efforts have enhanced our reputation as a quality program.

Providing vital services to other academic units was identified as another means of attracting new students. For years, the School's computer applications and introductory courses, required by many other disciplines, boosted our credit hour production figures, and provided a source of potential recruits. Recently, the School has added a minor, Information Technology, as an important new choice for majors in other disciplines. Courses designed for end users such as telecommunications, graphics, and office productivity, are included in this program. These courses have been positively received by the students, as enrollments are strong. Interest in the minor has been extremely high whenever we speak to groups of educators or students.

Retention

Attracting students into our program is the first hurdle, keeping them is the second. Over the last several years, the School has had two major curriculum objectives: (1) present a quality CIS program, meeting and exceeding the standards of the Computer Science Accreditation Board (CSAB) and the Joint Model for Information Science; and (2) retain students past the freshman year. Strategies for retention included: (1) Revision of the Freshman programming sequence, CS1 and CS2; (2) Expansion of tutoring services, and; (3) Faculty participation in the University's programs to promote student retention.

The CS1 and CS2 programming sequence is currently being revised to include the integration of "hands-on" structured lab activities for reinforcement of course concepts. This work is partially supported by the National Science Foundation. These additional labs provide that extra motivation, assistance, and encouragement needed the Freshman year. Free tutoring sessions, staffed by our student chapter of the ACM, are made available every quarter. In addition to the volunteer tutoring, the School supplements the volunteers by hiring more advanced students. The students who participate in the tutoring program become stake holders in the success of the School. This participation leads to a heightened sense of ownership and improves the entire educational climate.

Our faculty participate in the University's Mentoring Program that targets "at-risk" freshmen. A faculty advisor is assigned to each targeted student with the goal of developing a "mentor" relationship, providing support and encouragement. Similarly, "The First Time Student Program," designed for students who represent their family's first generation to attend college, is also supported by our faculty.

Conclusions

Our search for effective recruiting strategies began with a very basic motivation--survival. Successful strategies have resulted in an enrollment that is stable in the number of majors, and steadily increasing in the number of students taking the service courses.[Figure 2] Surprisingly, several desirable "side effects" have emerged as well. The faculty exhibit an increased awareness of and participation in the recruiting and retention of students. The vision of responsibility goes beyond course content. Improved curriculum and delivery of instruction have strengthened our program, in addition to promoting student retention. Cooperative relationships with faculty from feeder institutions are firmly established. Accessibility and open communication have promoted the image that our School is concerned with the best interest of the student, and that "best interest" includes a quality program.

Recruiting and retention in a changing field requires dynamic response to the changing needs and environment. The School is beginning to address an emerging need in information technology. We are offering new courses, providing community services and keeping communication channels open within the community. Recruiting and retention strategies in such an environment will require constant revision to keep pace with our changing discipline. In order to reflect this ever-changing definition, recruiting must be a continuous improvement process. The "next generation" of strategies will integrate telecommunications, taking advantage of the popularity of the information highway. Homepages and hypertext are just the beginning!

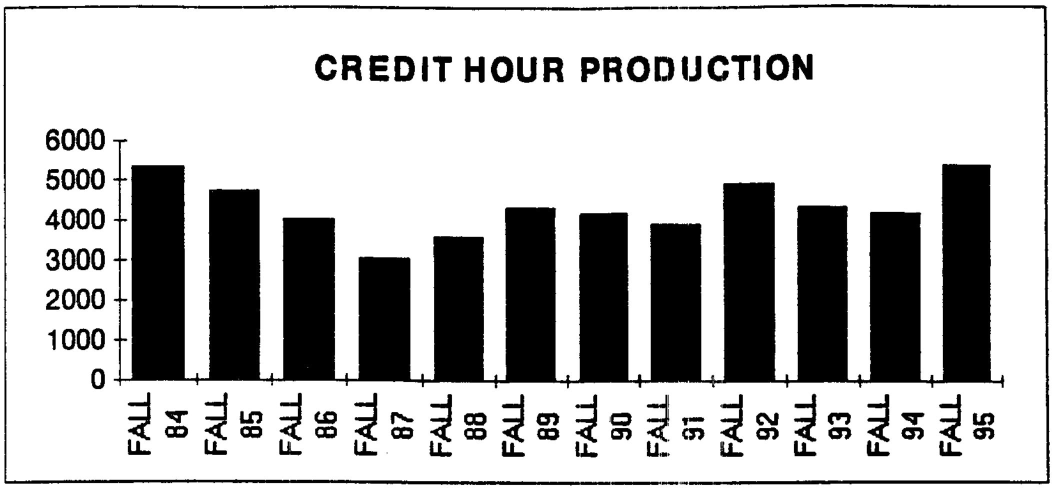


Figure 2

Bibliography

Marable, Leslie M. "The Fifty Hottest Jobs in America." MONEY; March 1995: 114-118.

Oberman, Dawn and Rhea Nagle. "The Job Market for the Class of '95". JOB CHOICES: 1995 ; College Placement Council, Inc.: 17-22.

U.S. Bureau of Labor Statistics. "The 1992-2005 Job Outlook in Brief." OCCUPATIONAL OUTLOOK QUARTERLY; Spring 1994: 10-21.

Zuckerman, Mortimer B., ed. "1995 Career Guide: Best Jobs for the Future." U.S. NEWS & WORLD REPORT, October 31, 1994: 86-120.

An Assessment of Workstation Ergonomics by Information Systems Management Personnel

Dr. Dennis LaBonty - Utah State University

Abstract

A survey was conducted of IS management personnel to assess the frequency of ergonomic practices and the importance of ergonomics in the workplace. The ergonomic assessment came from a questionnaire instrument administered to managers, engineers, and technicians representing different businesses. A rank order listing of statements about frequency of ergonomic practices and a rank order listing of statements about ergonomic importance was created using mean scores from Likert-like responses. This study reports what IS managers frequently do about ergonomic practices; and what they think is ergonomically important. The findings of this study can assist the IS curriculum. At a time when the information age is reaching full stride, at a time when businesses seek ways to reduce all costs, IS graduates can be prepared to help businesses towards safe, healthy, comfortable, and productive working environments.

Introduction

Are IS graduates prepared to provide ergonomic leadership for safe, healthy, comfortable, and productive work environments? This question comes at a time when offices are being computerized, and at a time when information processing jobs in automated offices are increasingly becoming health risks. These risks are costly and unproductive.

Healthy and productive work environments are things IS managers should have knowledge about. Armed with this knowledge, IS managers can exemplify an ideal ergonomic work environment and take positive actions that help businesses meet ergonomic challenges.

Approximately 40% of the workforce--about 40 million people--work at computer keyboards (Murphy, 1995). Additionally, it is estimated that 50 percent of America's workers are in jobs that are potentially susceptible to repetitive stress injures (Leavitt, 1994). When these estimates are combined, it becomes clear that computer-related injuries continue to consume a substantial amount of time and resources and handicap America's workforce and businesses. What is more, repetitive stress injuries represent half of all Worker Compensation costs, and these injuries have grown by over ten percent each year over the past ten years (Leavitt, 1994).

The stress and strain of the electronic office has produced many types of ailments. Some are physical like sore muscles, lower back pain, eye fatigue, and repetitive strain injuries (RSI)(Allie, 1994). These injuries result from physical and emotional stress or exhaustion. RSIs are also known as cumulative trauma disorders (CTDs).

CTDs are a class of musculoskeletal disorders involving harm to the tendons, bursa, or nerves of the hands, wrists, shoulders, neck, and back. Carpal tunnel syndrome (CTS)--a tendon, wrist injury--is a common CTD; tendinitis is another.

Ergonomic safety and comfort also include correction of sick buildings. Sick buildings are those with poor indoor air quality (IAQ), uncomfortable temperatures, and dim lighting. Sick buildings may be prone to high levels of static electricity.

Another computerized health issue is radiation from video display terminals (VDTs). VDTs emit a very low frequency electromagnetic field (VLF). The effects of VLF are unclear and unsettling. For example, in 1988 Kaiser Permanente reported a disturbing correlation between reproduction health problems among women who worked more than 20 hours per week in front of VDTs (Savage, 1993). The Environmental Protection Agency in 1990 found an "elevated risk" between cancer and electromagnetic fields, but the EPA reported in 1991 that the research was inconclusive and recommended further studies. Many experts today generally conclude that VDT workers are safe from VLF radiation but more research would be helpful. Leading monitor manufacturers have adopted the Swedish government MPR II standards for production of monitors with low emission of radiation--even though effects of VLFs remain questionable (Black, 1992).

Costs and injuries related to RSI are increasing. The Bureau of Labor Statistics (1994) reported that among finance, insurance, and real estate workers CTDs have risen tenfold over the past six years. Cessna Aircraft Company (1995) reported that CTDs accounted for 25 percent of the company's lost-time injuries.

Medical costs for CTDs are substantial. One carpal tunnel wrist surgery can cost as much as $13,000 (Carmody, 1995), while the average RSI claim is $43,000 according to a National Council on Compensation Insurance spokesman.

Ergonomics is the science of designing equipment to maximize worker productivity by reducing operator fatigue and discomfort while improving safety (Allie, 1994; Stamper, 1994). OSHA is involved in ergonomic safe offices as well as firms who have independently initiated ergonomic safety programs. Formal programs, like the one at the Sacramento Bee, advise computer users about computer ergonomics and safety. The effect of this program reduced the injury-related costs from $2.2 million in 1990 to $490,000 in 1994 (Peterson, 1995).

IS programs at the postsecondary level often are remiss about teaching ergonomic issues. This lack of knowledge is potentially costly to businesses and conceivably irresponsible to a computerized workforce of office workers.

IS programs should include information about ergonomics. Graduates from these programs should be prepared to help their companies provide safe, high-tech working environments. As businesses look for ways to reduce costs, IS managers can be positioned to help reduce medical costs, lost-time injuries, and improve productivity through ergonomic safety education.

IS managers should take the time to learn about their own working environments. A survey that assesses the IS workplace would benefit IS managers and IS graduates.

This paper proposes, if IS graduates are to gain knowledge about ergonomic practices and help businesses implement actions towards safe, healthy, comfortable, and productive workstations, they can begin by assessing current IS managers' work environments. Also, perceptions from current IS managers pertaining to important ergonomic practices would be helpful. Information from this survey can help shape and strengthen the IS curriculum.

Purpose of the Study

The purpose of this study was to determine how IS managers assessed their workstation environments related to ergonomic practices. This information can be used to shape the IS curriculum and to help IS managers and graduates mold safe and productive workplace environments. Also, this information can help managers become knowledgeable about ergonomic practices in their own work environments.

Methodology

Data Collection

The sample for this study was drawn from the attendees at a Management Information Systems seminar in 1995. The conference hosted over 250 attendees from business and industry as well as the academic community. Respondents were self-selecting with 97 of the questionnaires completed and usable.

A partial listing of corporations represented at the conference are listed alphabetically: Amalgamated Sugar Company, American First Credit Union, Bourns Integrated Technology, Campbell Scientific, First Security Bank, Hill Air Force Base, Hyclone Laboratories, IBM, Intel, Matrixx Marketing, Mineral Resources, Mountain Fuel Supply Company, Nucor Steel, Pacific Corp, Schreiber Foods, Thiokol Corporation.

These companies represent state, national, and international businesses. American First Credit Union is a state operated business while Bourns Network Incorporated is headquartered in California but is international in scope. These businesses sent IS managers, engineers, and technicians to a Management Information Systems seminar at Utah State University, Logan, Utah.

Data Analysis

The data was analyzed using the SPSS/PC+ statistical package for the IBM microcomputer. Mean and standard deviation calculations were used to rank the statements according to managers responses concerning frequency of practice and importance. Frequency of yes responses by managers were used to assess questions 20-23 and 34-39. The analysis of demographic and work-related information was calculated from percentages and mean averages.

Findings

Respondents represented different occupational status. The managerial executive, supervisor class made up 52 percent of all respondents. Engineering and technical respondents made up 20 percent of all respondents, administrative and office respondents were 12 percent, teachers and educators were 9 percent, and the remainder of respondents were people with varying backgrounds but interested in IS management. The average response to the length each respondent held their present job was 6.3 years, but they worked for the same company for 12.8 years. Managers averaged about 46 hours of work each week. Ninety-three percent were right handed. IS managers reported they spent 48.5 percent of their time in front of a VDT and that on the average they used VDTs for 12 years.

Participants were asked how frequently they used ergonomic equipment or actions. They responded to the statements using the following Likert-like scale 1 (Always) to5 (Never).

Table 1 shows the order of statement by frequency as per the mean scores. Column 1 represents the statement number in which the statement was listed on the survey instrument.

Table 1.
Frequency of Use Statements
Listed By Means Scores

Statement No.	Statement	Mean	Std. Dev
8.	Do you use electrical surge protection devices?	1.14	.56
6.	Do you use a caster-wheel chair?	1.25	.87
10	Do you use an adjustable angle computer monitor or video display terminal?	1.34	.94
16.	Do you arrange accessories or items--such as the mouse, telephone, reference documents, stapler--within easy reach?	1.61	.82
12.	Do you use a chair that adjusts including seat and backrest height?	2.09	1.54
9.	Do you use an adjustable angle keyboard	2.42	1.71
14.	Do you position wrists in a straight line position, not twisted up or down, or side to side?	2.63	1.13
13.	Do you position feet flat on floor or on a sturdy foot support during seated work?	2.75	1.35
15.	Do you re-adjust seated position throughout the day?	3.52	1.39
1.	Do you use stretching exercises throughout the day?	3.62	1.02
2.	Do you use a wrist rest, pad, or brace while typing?	3.67	1.69
3.	Do you use a telephone shoulder rest?	4.00	1.62
7.	Do you use a document holder during data entry?	4.22	1.13
4.	Do you use a glare or filter screen on video terminal display or computer screen?	4.37	1.25
11.	Do you use an adjustable computer desk?	4.66	.88
5.	Do you use a telephone head set?	4.82	.71

The four lowest mean score responses dealt with surge protection devices, caster-wheel chairs, adjustable angle VDTs, and accessories arranged for easy reach. The lower the mean score, the more frequently the statement's action was practiced. As most of the lowest mean scores were between 1 (always) and 2 (often) practice, it can be said that IS managers in this survey often use ergonomic equipment or practices such as using electrical surge protection devices, using caster-wheel chairs, using adjustable angle VDTs, and arranging accessories within easy reach.

The four highest mean score represent statements that were not practiced by IS managers. A high score was between 4 (rarely) and 5 (never) practiced. These statements consisted of using telephone head sets, using an adjustable computer desk, using VDT glare or filter screen, and using a document holder during data entry.

Statements that were ranked between the four highest and four lowest statements were using a wrist rest, using stretching exercises throughout the day, positioning feet flat on the floor during seated work, positioning wrists in a straight line, using an adjustable angle keyboard, and using a chair that adjusts the seat and backrest. These statements were rated as sometimes practiced by IS managers.

Participants were asked what their perceptions were about the importance of ergonomic equipment or practices. They used the following Likert-like scale 1 (extremely important) to 5 (extremely unimportant).

Table 2 depicts the order of statement by importance as per the mean scores.

Table 2.
Importance of Use Statements
Listed By Means Scores

Statement No.	Statement	Mean	Std. Dev
8.	Do you use electrical surge protection devices?	1.42	1.00
6.	Do you use a caster-wheel chair?	1.80	1.11
16.	Do you arrange accessories or items--such as the mouse, telephone, reference documents, stapler--within easy reach?	1.84	.94
10.	Do you use an adjustable angle computer monitor or video display terminal?	1.92	1.00
12.	Do you use a chair that adjusts including seat and backrest height?	2.10	1.15
14.	Do you position wrists in a straight line position, not twisted up or down, or side to side?	2.59	1.15
13.	Do you position feet flat on floor or on a sturdy foot support during seated work?	2.65	1.24
9.	Do you use an adjustable angle keyboard?	2.81	1.24
1.	Do you use stretching exercises throughout the day?	2.91	1.23
15.	Do you re-adjust seated position throughout the day?	3.09	1.28
2.	Do you use a wrist rest, pad, or brace while typing?	3.14	1.34
7.	Do you use a document holder during data entry?	3.40	1.20
4.	Do you use a glare or filter screen on video terminal display or computer screen?	3.44	1.38
3.	Do you use a telephone shoulder rest?	3.45	1.43
11.	Do you use an adjustable computer desk?	3.46	1.06
5.	Do you use a telephone head set?	4.12	1.21

The top four ranked statements as measured by their low mean scores were #8, #6 #16, and #10. These important statements were in the top four statements of the frequency of ergonomic practices too, however their order was slightly different. Likewise, the four least important statements were similar to the highest four statements of the less frequently practiced statements. The middle-most statements of the important group of statements were similar to the ranks of the frequently used statements.

A correlation to compare the rankings of the frequency and important statements was very high and positive. The Spearman Rank Correlation coefficient was .97, which means the order of the two groups of statements were similar.

Three statements on the survey sought information from the participants about the importance of the workplace environment such as temperature, air quality, and overall room lighting. The results were that overall room lighting was rated higher than the air quality, and air quality was rated higher than overall room temperature. Since the means average for these statements were between 1 and 2, it is apparent that IS managers perceive room temperature, air quality, and room lighting as very to extremely important.

Interestingly, IS managers' perceptions about room temperature, air quality, and room lighting were very important, but managers have little control over these environmental conditions in most working situations. As little as 25 percent of the respondents reported they could adjust the room lighting and only 24 percent reported they could adjust the room temperature. Eight percent of the respondents could adjust the air quality at their work places. Static electricity control devices were used by 26 percent of the respondents.

Table 3 shows the percent of respondents who had discomfort from their jobs.

Table 3.
Areas of Pain or Discomfort
Related to the Job in the Past Three Years

Area of Pain or Discomfort	% of Yes Responses
Back	46%
Neck	39
Shoulders	39
Wrists	28
Hands	28
Arms	9
Elbows	7

Back discomfort was reported from 46 percent of the respondents. Neck, shoulders, wrists, and hands were reported as discomforting by at least 30 percent of the respondents. Elbow and arm discomfort were listed by only 7 and 9 percent of the respondents.

Conclusions

Participants in the survey rated statements about the use of ergonomic equipment and practices based upon frequency of use and importance. The mean scores allowed the researcher to rank the two groups of statements. These two groups had a high, positive correlation coefficient and were similar. Therefore, IS managers frequently apply ergonomic practices to areas that they perceive important, and likewise, they do not frequently apply ergonomic practices to areas that they perceive as not important. If ergonomic practices are to improve, education about important ergonomic issues can help change perceptions thereby increasing the likelihood that they will be practiced.

Back, neck, and shoulder injuries were reported by 40 and 50 percent of the respondents; yet, respondents do not frequently "use stretching exercises throughout the day," "re-adjust seated position throughout the day," or "use a chair that adjusts including seat and backrest height." Even less frequently used were "use of a document holder during data entry" or "use of an adjustable computer desk." But IS managers in this study did rate statements like "use a chair that adjusts...," "re-adjust seated position...," or " use stretching exercises..." as very important.

IS managers have the opportunity and should re-adjust their seated position throughout the day as well as apply corrective stretching exercises while at their terminals. They reported in this study that they often do not. These actions could have positive effects in reducing stress, tension, and discomfort to the neck, back, and shoulders. Exercises that reduce stress and tension do not have to be rigorous physical workouts. Simple activities of rolling the shoulders and rotating the head in a circular pattern slowly are easy therapy. Up to this point it is uncertain if exercise prevents, reduces, or eliminates pain, but it is generally agreed that it does. What seems to help reduce or prevent discomfort should be practiced and not considered unimportant.

Thirty percent of the IS managers reported pain or discomfort to their hands and wrists. However, "use of a wrist rest, pad, or brace during data entry," and "use of stretching exercises throughout the day" were not frequently practiced and not rated as important. Wrist and hand care is not a top priority of the IS managers surveyed, even though these injuries are serious and copious. Some keyboards are applying new designs that claim to be more ergonomical to the hands and wrists, even though empirical evidence is not widespread. Those who suffer from RSI should attempt to use some of these newer designed keyboards.

Lighting, temperature, air quality, and static electricity were not commonly controlled by managers in this study. Buildings allow workers little opportunity to control their working environment because generally older buildings do not allow workers to adjust to comfortable work environments. If productivity is expected to increase however, businesses need to create flexible environments in buildings and permit workers some control over lighting, temperature, and air.

Arm and elbow injuries affected less than 10 percent of the respondents. Pain and discomfort to these body areas were not as prevalent as pain to other areas such as the neck, back, and shoulders. Even so, pain to these areas of the body need ergonomic attention.

Recommendations

1. A study should be conducted to determine the pain or discomfort to the eyes of people who spend large amounts of time in front of VDTs.

2. Research is needed that looks at other groups of computer users such as office workers, elementary and secondary students, and computer instructors. This type of study would be helpful in providing more information about areas where pain and discomfort are detected or provide remedies that reduce or prevent pain and discomfort.

3. More research is needed comparing gender in computerized work environments. Male and female workers, who spend the same number of hours at a computer keyboard, may report different information about injuries to body locations and severity of pain and discomfort. Also, a comparison of job position of men and women in computerized work environments would be informative.

4. A study about people who exercise and those who do not exercise should be conducted. People who exercise may report pain and discomfort differently from people who do not exercise. This research could be helpful for those who use computers often.

Learning more about ergonomics can be helpful to those who will spend much of their careers in computerized environments. IS curriculums should provide knowledge that will help IS managers become leaders in providing healthy, safe, and productive work places.

References

Allie, P. F. (1994). Ergonomics and the healthy office. Managing Office Technology, 39(2), p. 31.

Black, P. J. (1992). PC radiation: How to cut your risk. Business Week, 116, p. X.

Carmody, L. (1995). CAMA get it: CAMA Corp. is providing solutions in the area of ergonomics. Business People Magazine, 8(7), p. 40.

Fragala, G. (1995). Get more from your ergonomics training. Managing Office Technology, 40(11), pp. 45-46.

Guide your firm to a healthy workplace. Managing Office Technology, 38(6), pp. 28-30.

Hutchinson, R. (1995). OSHA rules could impact 120 million workers. Wichita Business Journal, 10(3), p. 4.

Leavitt, S. B. (1994). Casualties At The Office, (Leavitt Medical Communication, Glenview, IL).

Murphy, K. (1995, October, 9). Few standards exist on ergonomics; Approaches conflicting and choices in furniture abound. The Houston Chronicle, 3 STAR Edition, p. 3.

Peterson, R. (1995). Ergonomics and an office renovation; Bee invests in safety training, ergonomic offices. Editor & Publishing Magazine, p. 14P.

Reduce eyestrain, monitor glare, and increase productivity. Managing Office Technology, 38(6), p.14.

Savage, J. A. (1993). Are computer terminals zapping workers' health? Business and Society Review, No. 84, pp. 41-43.

Stamper, D. A. (1994). Business Data Communications (4th ed.). Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

Using Visual Basic in a First Programming Course

Cathy Bishop-Clark, Associate Professor
Systems Analysis, Miami University, Ohio

Abstract

Traditionally, introductory programming courses were taught in BASIC or Pascal. It took student several weeks before they could create even a simple program. Visual Basic allows students to create highly visual, relatively sophisticated programs with little experience. It is an exciting and engaging language. During the fall of 1995, Visual Basic was used on an experimental basis in two sections of an introduction to computer programming course. The course introduced students to fundamental programming concepts such as sequence, selection, iteration, file processing, and arrays. Students were asked to comment throughout the course and filled out a final evaluation of their experience. The instructor also closely monitored student's understanding of the fundamental programming concepts. Overall, both student and teacher evaluations were overwhelmingly positive. Visual Basic offers new opportunities and promise to beginning programming courses.

Using Visual Basic in a First Programming Course

An introductory programming course has many objectives. One of the objectives of such a course is to excite students about computing and programming. For non-majors, the course may be the only formal exposure they have to programming. For majors and potential majors, the introductory programming course creates their first impression of programming and is often the deciding factor on whether to major in computing or information systems.

Traditionally, the introductory programming course was taught in BASIC or Pascal. It took students several weeks before they could create even a simple program. The interface was largely textual and for the most part, beginning programs were boring. It took at least a week to learn how to write a program to compute the area of a circle. Introductory programming courses typically have a low retention rates and many students leave the course hoping to never take a programming class again.

Visual Basic (VB) allows student to create highly visual, relatively sophisticated programs with little experience. It is an exciting and engaging language. Although introduced only recently, Visual Basic is very popular. Because Visual Basic can be used easily in most Microsoft applications, the language is becoming ubiquitous. In the foreseeable future, the development of the language seems assured. Until recently, most of the books available for Visual Basic were reference type; however, several introductory textbooks appropriate for an Introductory programming class (Coburn, 1995; Schneider, 1995; Zak, 1995) are currently available. There is also an extensive on-line help system to help novice users. Just as C/C++ has replaced Pascal in most universities, it appears to be just a matter of time before Visual Basic will replace other versions of Basic--especially in introductory courses.

This paper describes a pilot study which involved using Visual Basic in 2 sections of an introductory programming class. The course is a popular one and approximately 15 sections of the class are offered accross three campuses of the University. After a brief literature review, the course and course requirements are described. Student and instructor reactions to Visual Basic in a first programming course are discussed. Based on the experience of teaching the two pilot sections in Visual Basic, the author concludes that Visual Basic is an excellent choice for an introductory programming course.

Literature Review of Using VB in an Introductory Classroom

Although the language is quite new, several educators report on using Visual Basic in the classroom. Davey and Tatnall (1995) report of a case study in two Australian universities. Over two semesters, the authors taught several sections of a course in two different Universities. All of the students had previous programming experience. The authors administered questionnaires at the end of the semester, and kept a log of the teachers experience. They point out that <u>initially</u> previous experience is not directly transferable to VB which creates a challenging situation for both students and instructors. They also report that students found Visual Basic exciting, engaging and fun!

Cox and Clark (1994) suggest that Visual Basic be the language of choice for service modules, user interface courses, and some applications development projects. They explain that Visual Basic is an excellent choice for the language used in service type courses. Visual Basic provides easy syntax, the environment supports rapid implementation, the product student create is exciting, and incremental development is easy and highly visual. They explain that this is the kind of language used to inspire today's "cellular phone generation.". Both articles make some excellent points about when to use and when not to use Visual Basic as the language to teach programming concepts.

Overview of the Course

SAN 163 (Introduction to Computer Systems and Programming) is a liberal educational course at a medium sized midwestern liberal arts college. The technical objectives of the course are to develop the students understanding of basic computing concepts and terminology; to perform the coding, documentation, testing, and implementation of well-structure computer programs; and to learn to analyze a problem and apply algorithmic reasoning to design a well-structured robust computer program. While the course has some very concrete technical objectives, it is also considered a liberal education foundation course. The activities of the course are designed to develop the student's ability to think critically, to reflect on their learning, to improve their ability to work with others, and to understand the context of a problem and a solution.

There are primarily two groups of students in the class: systems analysis majors and non-majors. The systems analysis majors are taking the course because they have no previous programming experience. This course is a prerequisite to the first required programming class in the curriculum (in C++). Students are expected to master fundamental programming concepts such as sequence, selection, iteration, arrays, and files.

The second group of students are non-majors and are taking the class to fulfill one of the liberal education requirements of the university. For many, it is the only exposure they will have to computer programming. This class satisfies the "mathematics, formal reasoning and technology" requirement of the university. The language used in the course must be one that satisfies the objectives for both groups of students.

The topics of the course include both computing concepts and vocabulary and programming. Two books are required--one for programming (Schneider, 1995) and one for computing concepts (Shelly, Cashmand and Waggoner, 1993). The computing concepts is covered primarily by outside reading assignments. The majority of class time is devoted toward programming concepts. Table 1 shows weekly agenda for the course.

Table 1: Topical Outline

Week	Topic
1	Intro to the Course, Liberal Education, Computers
2	Intro to Problem Solving, Visual Basic
3-4	Fundamentals of Programming
5-6	Selection Structures
7-8	Repetition Structures
9-10	Procedures
11-12	Arrays
13	Sorting and Searching
14-15	Sequential Files

Course Requirements

The course is quite demanding and requirements include programming assignments, writing assignments, in class laboratories, quizzes and exams.

Programming Assignments: Approximately 10 programming assignments are given throughout the class. The programming assignments are to be completed independently. The first assignment involves simply creating user interfaces. The students progress to writing programs that work with files and arrays. The final project (programming assignment 10) was completed in pairs. Students identified, developed and coded their own application. The only restriction was that the final project had to incorporate most of the ideas of the class. Most of the students created projects which included arrays and files. Final projects were demonstrated to other students in a poster session form on the last day of class. Several students created a game such as black jack, hang-man, and tic-tac-toe. Others created a telephone directory system and a magazine ordering system.

Writing Assignments: Because the course is a liberal education course, approximately five 2-page writing assignments are given throughout the semester. Two of the writing assignments (one very early in the term and one during the last week of the term) asked the students to assess Visual Basic and programming. They were asked about what they enjoyed and what they found frustrating.

Laboratories: Approximately 1/3 of the class time is spent in laboratories. Laboratories provide the student with a hands-on experience in a supervised setting. Students were always assigned a lab partner, although they usually worked on their own machine. Students were strongly encouraged to help their lab partner. In fact, on several lab assessments, students were asked to comment on how they helped their partner learn! Students turned in a short report for each laboratory experience. The report often included a short program.

Quizzes: Regular quizzes were given throughout the course on the computing concepts. These concepts were not discussed in the course unless the student requested to review a certain topic.

Exams: Two exams and a comprehensive final were given throughout the class. Table 2 shows the breakdown for determining the students final grade.

Table 2

Component	Percent
Exams	25%
Programming Assignments	25%
Writing Assignments	10%
Quizzes	10%
Labs	5%
Final Exam	25%

Student Reactions to Visual Basic

Students were asked to give their reactions to the language throughout the class. Specifically, two of their writing assignments asked them to comment on Visual Basic. The first writing assignment was completed the third week of the semester and last was completed the final week of a 16-week semester.

Overall, the evaluations were overwhelmingly positive. Students commented that they found the language exciting and interesting. Many also commented that they found Visual Basic easy to learn and that the visual interface made the concepts easier to understand. Table 3 illustrates the kinds of the written comments made at the end of the semester.

Table 3

Student Comments
As far as learning to write programs goes, I think Visual Basic not only made it interesting , but also easier to
The visual programming concept has made it extremely enjoyable for me and I don't know if I could have understood the concepts otherwise.
While I do not find it (Visual Basic) especially easy, I do find it very exciting.
In Visual Basic, the interface is easy to create and rewards your hard written programs with a proper presentation, making all your work more worthwhile.
Visual Basic has given me the feeling of controlling computers.
Visual Basic gives the programmer a chance to be creative.

Not all student comments were positive. A few students commented that they disliked the primer (a scaled down VB package that came with the book). Some students commented that they lose themselves on the screen and can't find where they need to be because of the multiple windows. Several other students disliked the specific ways Visual Basic handles particular tasks--such as the save routine.

In addition to qualitative feedback, on the last day of class the students were given a short likert-type survey which asked each student to respond to each statement with a strongly agree, agree, neutral, disagree, or strongly disagree. For the purposes of presentation, the strongly agree/agree and strongly disagree/disagree are combined. Table 4 indicates the results are consistent with the verbal descriptions--overwhelmingly positive.

Instructor Reactions to Visual Basic

As the instructor of the course, I kept a journal thoughout the class where I commented on my expectations and reactions. Initially, I felt the language would be an excellent choice for our non-majors but I was concerned about whether the language would be a good one to teach the fundamentals of programming for our majors.

Visual Basic provides an opportunity to focus on the package instead of programming. Such a focus would compromise the emphasis on structure and logic. However, with appropriate instruction the emphasis can be on fundamental logic structures. In our class, we reviewed only the very basic features of the interface. Instead, the majority of time was spent on the fundamental logic structures. Discussions with the instructor of the next course indicate that the Visual Basic students were just as prepared as Qbasic students for C++.

Programming in Visual Basic requires a first step that is not required of traditional basic--the design of the interface It seemed that Visual Basic facilitated the identification of inputs and outputs to a program via the design of the interface. The design of the interface forced the students to initially identify the inputs and the outputs of the program before they began coding. As the semester progressed students designed very elaborate interfaces. Interestingly, while many students created elaborate interfaces, they were also poor interfaces. They were often busy and difficult to understand.

Table 4

Question	Agree	Neutral	Disagree
1. Writing computer programs is fun.	18	4	1
2. I enjoyed learning the programming language we used in this class.	22	1	0
3. I enjoyed creating the user interface used in the program	20	3	0
4. Because of this course, I will consider majoring in Systems Analysis.	12	5	6
5. I enjoyed thinking through the logic required to write computer program.	17	6	0
6. I would take another computer programming class.	18	2	3
7. I am sure I could learn another computer language	20	1	2
8. This course made me excited about computing.	18	4	1
9. I better understand how computer systems work.	20	3	0
10. I feel I am in control of the computer system	14	8	1
11. I understand how programs follow a sequential nature from beginning to end.	22	1	0
12. I understand fundamental programming concepts	22	1	0
13. I understand arrays.	21	2	0
14. I understand selection structures (if-then-else, case)	20	3	0
15. I understand looping structures (while, for)	21	2	0
16. I understand how programs can be broken down into modules	16	7	0

Students have the same kinds of problems in Visual Basic as they did in other forms of BASIC. For instance, students initially struggle with arrays and struggle with the various forms of looping. Student comments indicate that they found debugging the most frustrating. There were far more similarities between programming in Visual Basic and QBASIC than differences.

Without question, the single most exciting aspect of switching to Visual Basic was the student enthusiasm. Students found Visual Basic exciting and this excitement began very early in the class with the introduction to creating interfaces. As early as the second week students were designing interfaces with buttons, pictures and text boxes. This enthusiasm is best illustrated with the presentation of the final projects. Recall, the students identified their own final project and over half of the students went beyond the requirements of the final project. They learned additional information outside of class on programming in Visual Basic! In the end, they were proud of their final project and were eager to share the programming behind the application with their classmates.

Conclusion

Student and instructor reactions indicate that VB may be an appropriate first language for courses whose primary objective is to introduce students to computing and fundamental programming concepts. However, there are instances where VB may not be the best choice for a language. Both Davey and Tathall (1994) and Cox and Clark (1995) warn against using Visual Basic in certain situations. Cox and Clark (1995) warn against using VB to teach modularization, data structures, and abstract data types--other language such as C++ allow for far better features to support these. Recall that Davey and Tathall suggest that VB may not be the best choice for a first programming language because of the complex mental model it requires and because of the multiple concepts involved--BASIC programming, event-driven code, object oriented techniques. The authors raise a valid concern; however, VB can be taught in a way that emphasizes fundamental programming concepts and de-emphasizes event-driven and object-oriented concepts. By introducing students to only a few features of VB, we allow the students to focus on the logic structures behind the interface. By creating test questions, assignments, and lab projects which focused on the fundamental logic structures, one can create a course that does focus on traditional programming logic.

In summary, based on this experience, Visual Basic appears to be an excellent choice as a language for a beginning programming course--a course whose focus is on teaching the fundamental programming constructs of sequence, selection, iteration, arrays, and files. Visual Basic creates an excellent first impression of programming. It is a good language for beginners. The syntax for Visual Basic is easy and the environment supports quick implementation. This gives the beginning student a sense of achievement and a feeling of having control--which will hopefully translate into additional confidence. The final product they create is one that they enjoy showing their family and friends. The marketability of students is enhanced if they have Visual Basic skills. A quick look at the classified adds continues to confirm the demand for VB skills. This is especially important at two year institutions where students often enter industry with less computing experience; however, it may help the coop and summer intern possibilities of all students.

Although difficult to measure, it appears that Visual Basic helped retention of students. Out of 24 students, only one dropped the class after the fourth week of the class. In addition to retaining students, it appears that the course caused at least some students to consider switching their major. Based on this first impression, several students who began the class "undecided" decided to enroll in the next course (C++). Several students officially changed their major to Systems Analysis during the course. Finally and perhaps most important, most students found VB exciting and interesting to the point where they learned the language beyond the requirements of the class

Bibliography

Coburn, E. (1995) Visual Basic Made Easy, PWS Publishing Company, Boston, Massachusetts.

Cox, K.R., Clark, D. (1994) Computing Modules that Empower Students. Computers and Education, 23(4), pp. 277-284.

Davey, B. and Tathall (1994), A. Introducing Object Environments: Cognitive Difficulties. Proceedings; Software Education Conference, pp. 128-133.

Schneider, David I. (1995) An Introduction to Programming using Visual Basic, Prentice Hall, Englewood Cliffs, New Jersey.

Shelly, Cashmand and Waggoner (1993) Essential Computer Concepts, Boyde and Fraser Publishing.

Webb, B.R. (1995) Teaching Usability to Visual Basic Programmers: A Case Study. Software Engineering in Higher Education, Computational Mechanics Publications, Southampton Boston, pp. 209-216.

Zak, D. (1995). Programming with Microsoft Visual Basic 2.0/3.0 for Windows, Course Technology, Cambridge, MA.

COBOL - FUTURE DIRECTIONS

Alden C. Lorents, Northern Arizona University, Flagstaff AZ
520-523-3510 alden.lorents@nau.edu

Abstract

COBOL has been a controversial topic in CIS curriculum discussions for many years, and continues to be controversial to the point that many curriculums are downplaying its use, and eliminating any reference to it in course names and descriptions. Discussions regarding COBOL versus other languages causes concern among students over the credibility of the program because they are still being taught an old language. The intent of this paper to look at where COBOL is today in terms of industry use, language standards, curriculum changes, and to look at some potential directions of the language over the next ten years.

Introduction

COBOL continues to remain the predominant language behind roughly eighty percent of the enterprise wide systems in the larger corporate organizations across the US today. The buildup of the these systems started in the mid 1960's when COBOL was introduced, and continues today. The estimates vary widely on the amount of code that is currently in active production. Some of the estimates are shown in Figure 1. [10] [1]

Many organizations have worked on moving some of their enterprise-wide applications to client-server based systems using tools like Visual Basic and Powerbuilder with databases such as Oracle and Sybase. However, these tools do not have the scalability to handle larger systems used in banks, insurance companies and larger corporate organizations. "Neither object-oriented database technology nor distributed UNIX-based databases will replace mainframe database management systems any time soon" [10] according to users at a recent DB2 users group meeting. The primary host language supporting these systems is COBOL. A large utility just pulled the plug on a $23 million project that was to use Oracle and Object development tools from Oracle. The project has been switched to DB2 using COBOL II for the backend, and Visual Basic for the client stations.[14] Most organizations with large COBOL based systems do not see themselves phasing out these systems within the next ten to fifteen years. The big question is: What will be the transition path for these legacy systems over the coming years?. The vendors who have a big stake in the future of COBOL are hoping it will continue to be COBOL. These vendors have invested millions of dollars into developing Object Oriented COBOL with GUI support and other modifications to conform to and support the 97 Standard. Vendors include Hitachi, IBM, Ryan-McFarland, Computer Associates, and Micro Focus.

COBOL '97

The draft standard for COBOL '97 is in its final stages of review. The proposal includes the implementation of Object oriented programming capabilities found in C++ and Smalltalk. These include inheritance, polymorphism, and encapsulation. A comparison of the various feature are presented in Figure 2.

Other features expected to be included in the COBOL 97 standard are:

1) Program Pointers - A new data type which contains the address of a program. Useful with APIs that call back to a program to handle an event.
2) Data Pointers - a new data type which contains the address of a data item. Useful when interfacing to other languages, especially C.
3) Recursive programs - allows a program to call itself, directly or indirectly.
4) Expanded arithmetic, bit and large character sets.
5) Automatic expanded tables and dynamic file allocation.
6) Exit Perform, Exit Perform Cycle, Exit Paragraph, Exit Section.
7) Year 2000 enhancements.
8) A common method for exception handling and the management of errors
9) Bit string handling
10) Expanded compiler directives
11) Support for double byte character sets
12) New numeric types (Native binary, Floating point of the machine, 31 digit fixed point)

Object Oriented Cobol

Object Oriented COBOL - Class Program Structure. A single set of source code, called a class program, is used to contain all the class data, class methods, the object instance data and the object methods for a single class and the object created by this class. A diagram of this structure is shown in Figure 3. An example of COBOL Object code that demonstrates this structure is shown in figure 4.

A Class is like a template that is used to create objects. In object oriented COBOL, a class is implemented as a program that consists of a set of nested programs. When the program is run, the class is loaded and the run time system creates a class object which represent the class. Instances of this class can be created from this class.

A traditional COBOL program consists of sections and paragraphs. A class program consists of methods. A method is like a small program that can access locally defined data and reference instance data defined in the object-storage section. This is a new data division section that allows data that is declared within it to be inherited. With inheritance, a class (program) can inherit the state and behavior of another class.

An instance of a class, called an object, is created by sending it the "new" message. This is done by using the INVOKE verb. The INVOKE process passes parameters much like the Call statement. The sample program that follows shows a partial example of the class "checkaccount". It is actually a subclass of the class account and inherits behavior form the account class. The method "withdraw" is a procedure to subtract the amount withdrawn from the balance.

Another approach to using OO COBOL is to use COBOL as a Class. You can compile a procedural program into a class containing methods for each of the sections and paragraphs of the original program. Working-storage is compiled to object-storage and PERFORM statements are compiled to INVOKE statements. Once the program has been compiled to a class, it can be re-used. "All you need to do is write a modified section or paragraph, add any new sections or paragraphs as necessary, and add any additional working-storage items. Then, you compile this program, specifying an appropriate directive to the compiler together with name of the class it is to inherit (the original procedural program) and it is compiled into a subclass." [16]

Object COBOL Proponents

Will object COBOL be accepted in the market place? What are the benefits of Object COBOL? Does object orientation have a place in the future of enterprise-wide, mission-critical application of larger organizations? These are very good questions that may not have good answers in the next few years. Some of the justification for object COBOL are embedded in the following topics:

1) Existing COBOL base (Staff and code). If a decision is made my an application development group to start using object oriented methods, and they already have a large investment in COBOL code and in COBOL developers, it may be easier to make the transition with a language that has some familiarity. Training COBOL developers to switch to new languages such as C, C++, and Smalltalk is very expensive. Also switching to C and C++ is not an acceptable alternative for most existing COBOL systems. This would lead eventually to legacy systems built in a language that is more difficult to maintain than COBOL.

2) Language Standard. COBOL's success over the years has been due to the fact that it is based on a standard that is supported by a large following that includes manufacturers, users, and governments. A language based on a standard that is widely accepted has the advantages of: a) portability across various platforms, and b) support by a large group of developers who know the language. Organizations that uses Visual Basic or Powerbuilder deal with the problems of a short supply of developers, and a commitment to certain platforms. "OO COBOL satisfies the need for portability; even though the X3J4 standards group moves slowly; it does support an international standard---and it is a standard that is more stable and more widely supported than *any* of the other programming languages." [2]

3) Enterprise Systems: Most of the legacy COBOL is supporting enterprise-wide systems that are massive systems in terms of transaction volumes, file sizes, and the numbers of work-stations being supported. Most efforts to transition systems into new systems with various tools such as Visual Basic, Powerbuilder, and various client-server based development tools, are okay with smaller systems, but they are not scaleable to larger systems. A recent comment by Oracle executive relative to the Duke Power company $23 million object oriented project that was halted stated that " The object-oriented methods and tools proved not to be scaleable to a problem of this size and complexity".
Duke is continuing with the project using DB2, COBOL II, and Visual Basic to support the front end GUI for the 1200 work stations. [14]

Legacy System Transition and Reusability. One of the possibilities in the transition of legacy systems is to use parts of the existing code in the development of new systems. This can be done by including that code in an object wrapper. Although this can also be accomplished by other object oriented languages, it may make more sense to do it in the same language.

Object COBOL Skeptics

There is also various skepticism regarding the future of Object COBOL in addition to the traditional skepticism about COBOL in general. Some of the skepticism around object COBOL is related to the following areas:

1) Many organizations are moving their legacy systems to client server based systems where the corporate wide databases remain on larger server platforms leaving the development activities to departmental computing staffs. These staffs will be using Oracle Case tools, Visual Basic and Powerbuilder to develop the front end applications and reporting needs.

2) Object oriented techniques have been developed primarily in the scientific community where applications may be more appropriate. There is some concern that object oriented development may not be as appropriate in problems where business rules are very dynamic. However, the dynamic binding features of OO COBOL should be able to alleviate some of these problems.

3) Moving new technologies into large legacy systems is a slow process and will take a number of years. Acceptance of Object COBOL will depend somewhat on the acceptance of object development in general for these applications. "The slow start object-oriented COBOL has made in the business consciousness parallels the progress of other object-oriented programming languages. Right now, everyone is skeptical of OO in general." [5]

4) COBOL has a reputation of being a stodgy language that has been slow to respond to changing development methodologies and needs. Managers have not kept up with the changes that COBOL 85 and COBOL 97 brings to the language. The language has always been looked at as a second class citizen compared to any language used in the computer sciences. Part of this comes from the fact that COBOL is a business language, and it is used in business applications. It is never used in scientific applications. "One thing all the experts agree on is that COBOL will always remain a business applications language, but it has since jettisoned its stodgy, character-based persona for the more colorful personality of a 'modern' language." [5]

Supply and Demand for Developers

Most reports regarding careers for the next ten years rank computer analysts and systems analysts in the top ten careers in terms of demand. There is no question that the demand will continue to be high for most positions in the various areas of information systems development. Key development preparation areas that will help graduates in the next few years include: 1) database design (relational, IMS, IDMS), 2) SQL, 3) GUI design, 4) network, telecommunication, and internet/intranet design, 5) language preparation (Visual Basic, C/C++, COBOL/Object COBOL), 6) CASE tool orientation (IEF, SAP, Oracle), 7) System modeling including object oriented modeling. The amount of preparation in some of these areas depends on the interest of the individuals and the markets they want to go into. It is more difficult to prepare graduates today, because
the variety of jobs has increased dramatically. Larger programs can provide multiple tracks. Smaller programs have to decide on their niche areas. "One solution is to form a major with two tracks (1) a PC track that omits COBOL and (2) a mainframe track that places heavy emphasis on COBOL" [17]. COBOL will continue to be a high demand area for individuals who are interested in going to work for larger organizations, or for consulting companies that do work for these organizations. There is some concern that there will be a shortage of COBOL developers around the year 2000. This could come about for a number of reasons: 1) Demand will be high in various technology areas that will tend to draw from the existing pool of COBOL developers, 2) There are a number of COBOL developers retiring each year, 3) Demand will increase due to the increased maintenance surrounding date routines, 4) Object COBOL development could increase the activity in migrating legacy systems into a more manageable environment by gradually moving some of the code into an object environment, and 5) There is less emphasis today to prepare graduates with a COBOL background because other technologies are more popular.

A big factor that will affect the demand for COBOL based developers in the future will be how fast and to what extent Object COBOL is accepted in the market place. If Object COBOL is accepted, it could be a major enabler for larger organizations to gradually migrate their enterprise-wide systems into new systems in the next decade. In any event the large COBOL base will not go away for many years. The investment in this base is very large and the solutions to migrate these investments into new systems come about very slowly. It also appears that the COBOL language will continue to be upgraded to meet the changing demands necessary to support large enterprise-wide system development and migration. A cartoon a few year back depicts one developer asking another developer: "What will the development language of the year 2000 look like?" "I don't know", said the other developer, "but it will be called COBOL"

Figure 1

Lines of production COBOL code in use (Worldwide):
Over 150 billion LOC.
COBOL programmers add about 5 billion lines each year.
Datapro Information Services Group

Percentage of all code that is written in COBOL:
80%
Full Time COBOL programmers (Worldwide):
Over 2 million
Projected 1996 market for COBOL development:
$3 billion
COBOL programmers who still work at terminals:
40%
International Data Corp.; Gartner Group Inc.

Banking Industry production COBOL code:
Over 57 billion LOC
American Banker Tower Group survey October 9, 1995

COBOL desktop development revenues for 98:
$176 million
Growth rate is about 15% per year.
Percentage of all application staffs using COBOL
43%
35% of all companies use COBOL for 67% of their applications.
International Data Corp.

Figure 2

Feature	**Smalltalk**	**C++**	**COBOL**
Pure Object Orientation	Yes	Hybrid	Hybrid
Inheritance	Single	Multiple	Multiple
Encapsulation	Total	Limited	Total
Polymorphism	Yes	Yes	Yes
Classes as Objects	Yes	No	Yes
Type Checking	No	Yes	Yes
Class Libraries	Excellent	Good	Limited
Persistent Objects	Yes	No	Simple

* Reference [1]

Figure 3

```
Class-id. ClassName.
    Class-Object.
        Class Data
        Method-id. "ClassMethod".
        Method Data and Method Processing
        end Method "ClassMethod"
    end class-object.
    Object.
        Object Data
        Method-id. "ObjectMethod".
        Method Data and Method Processing
        end Method "ObjectMethod"
    end object
end class ClassName.
```

Figure 4

SAMPLE OBJECT COBOL PROGRAM [Micro Focus Object COBOL Sample Programs]

```
class-id. CheckAccount
     data is protected
     inherits from Account with data.

  object section.
  class-control.
     Account is class "account"
     CheckAccount is class "caccount".

 working-storage section.
     copy "accinfo.cpy".
 class-object.
 object-storage section.
*--------------------------------------------------------------
* Set the first account number for Check accounts.
*--------------------------------------------------------------
 Method-id. "getFirstAccountNumber".

 linkage section.
     01  lnkNumber               pic 9(8).
 procedure division returning lnkNumber.
*----C0001. Set up account numbering for checking accounts.
     move 10000000 to lnkNumber
     exit method.
 end method "getFirstAccountNumber".
 end class-object.

 Object.
 object-storage section.

     01  permittedOverdraft         pic S9(10)v99 comp-5 value 0.
*--------------------------------------------------------------
* Implementation of "withdraw" which allows for overdrafts.
*--------------------------------------------------------------

 method-id. "withdraw".

 working-storage section.
     01 workingValue           pic S9(10)v99.
     01 errorCode              pic x(4) comp-5 value
                               debitExceedsOverdraft.
 linkage section.
     01 lsAmount               pic 9(10)v99.

 procedure division using lsAmount.
*----C010. Withdraw lsAmount from this account
     subtract lsAmount from balance giving workingValue
     if workingValue < permittedOverdraft
         invoke self "raiseException" using errorCode
     else
         move workingValue to balance
     end-if
    exit method.
 end method "withdraw".
```

References

1. Saade, Henry and Ann Wallace, "Cobol "97: A Status Report, *Dr. Dobb's Journal*, October, 1995.
2. Yourdon, Ed, "Micro Focus's Object-Oriented Cobol", *Application Development Strategies*, February, 1995.
3. Sutherland, Jeff, "Smalltalk, C++, and OO Cobol: the Good, the Bad and the Ugly", *Object Magazine*.
4. Kizior, Ronald J. "The Resurgence of COBOL: Considerations for the Future", *Journal of Information Systems Education*, Fall, 1994.
5. Sarath, Patrice, "Object Oriented Cobol - Adding a dash of panache to the world's most boring computer language", *Access to Wang*, September, 1995.
6. Schmidt, Jennifer, OO Cobol: "Applying Existing Skills to Exploit a Reuse Option", *Application Development Trends*, February, 1995.
7. Carthcart, Mark, Object Technology for MVS, System/390 White Papers, April, 1995.
8. Obin, Raymond, *Object-Orientation: An Introduction for Cobol Programmers*, Micro Focus Press, 2nd ed. 1995.
9. Tracey, Brian, "Legacy Systems Are Still Very Much With Us", *American Banker*, October 9, 1995.
10. Busse, Torsten, "DB2 for MVS: Alive and Thriving", *ComputerWorld*, November 6, 1995.
11. Bayer, David, "C/C++ I love its portability, I hate its complexity" *ComputerWorld*, October 23, 1995.
12. Arranga, Edmund C., "Teaching Object-Oriented COBOL - A Report from the Classroom", Micro Focus *AGP Journal*, Nov/Dec, 1995.
13. Hurwitz, Judith, "OO COBOL: Panacea or Paper Tiger?", *DBMF*, August, 1995.
14. Hoffman, Thomas and Julia King, "Utility unplugs object project", *ComputerWorld*, February 26, 1996.
15. Grehan, Rick, "Object-Oriented COBOL", *Byte*, September, 1994
16. Archbell, Ian, COBOL As a Class - An Evolutionary Approach to OO, *Compilations*, issue-95-2.
17. Sass, C. J., "Job Market Changes and its Effect on Curriculum Issues in IS in the 90's", *IBS Computing Quarterly*, Winter, 1995.
18. Marquis, Gerald P., "Intoducing Object-Based Concepts in a COBOL Course", *Journal of Information Systems Education*, Winter, 95-96.

Integrating Ethics into the Business Curriculum

Richard Glass
Bryant College

ABSTRACT: The need to incorporate ethics into the information systems curriculum has been recognized by both the academic and professional communities for some time. The purpose of this workshop is to address how we as instructors can incorporate ethics into our courses in order to have a positive influence on the ethical conduct of our students. The main theme that will be presented in the workshop is that in order to raise student's consciousness about ethical issues, students must be challenged to experience conscious ethical conflicts and to incorporate their own values into solving ethical problems. To effectively achieve this goal, students must be faced with a number of different ethical dilemmas provided in several of their courses. Methodologies and approaches will be discussed in the workshop that may be tailored to specific needs of any information systems course.

OUTLINE

INTRODUCTION

1. Overview of the workshop
    - what the workshop isn't
    - what the work shop is
2. What is ethics and why should we teach it?
3. Why is ethics so hard to teach?
4. Can we teach moral values?
5. What can we teach?
6. What approach should we take?

A MODEL FOR TEACHING INFORMATION ETHICS

1. Objectives for teaching ethics
2. A model for teaching ethics
3. Normative Ethical Theories
    - What are they?
    - How can we use them in our classes?
4. Overview of how we may teach ethics

BREAK

METHODOLOGIES FOR TEACHING BUSINESS ETHICS

1. Issues in ethics
2. Popular techniques for teaching ethics - pros and cons
3. A multimethod technique based on role playing

QUESTION AND ANSWER SESSION

AUTHOR'S BIOGRAPHY

Richard Glass is an Assistant Professor of Computer Information Systems at Bryant College in Smithfield, RI. He holds an MBA from the University of Western Ontario and a PhD in MIS from Concordia University. He has published on the topic of ethics in the Journal of Business Ethics, Journal of Computer Information Systems and the Journal of Information Systems Education. He is a Fellow of the John Hazen White Sr. Center for Ethics and Public Service and is involved in providing ethics seminars and training for the public sector. Dr. Glass currently serves as Associate Editor of the Journal of Information Systems Education. He is a director of the Data Processing Management Association (DPMA) Special Interest Group on Education (EDSIG). His research interests include information ethics, IS education and computer assisted decision making.

Alternative Models for the Description of CPU Organization and Operation in Modern Computers

Irv Englander, Ph.D.
Computer Information Systems
Bentley College
175 Forest Street, Waltham MA 02154
(617)-891-2585, e-mail: ienglander@bentley.edu

Abstract

A survey of the hardware modules that appear in fifteen current introductory Information Technology textbooks shows that the model generally used to describe the organization of the CPU in introductory IT courses is outdated and unrepresentative of modern CPU organizations. This paper presents a general model of the CPU that is more suitable for the description of modern computers. It also shows how the new model can naturally improve the student's understanding of CPU operation.

Description of the Problem

Nearly all introductory Information Technology courses include a module that provides a description of the hardware organization and operation of computer systems. An understanding of computer hardware has increased in importance to students in recent years with the near- universal access to personal computers. We surveyed fifteen popular current textbooks for this course, with titles such as *Introduction to Management Information Systems* or *Using Computers in Business*, to determine the focus, accuracy, depth, and currency of their hardware module coverage. Our purpose was to determine if the coverage was appropriate and sufficient to meet the needs of non-IT business and liberal arts majors, and whether it provided an adequate foundation for later discussion of computer architecture for IT and IS majors.

The objectives and emphases of the surveyed textbooks range widely, from those books that concentrate almost entirely on the use of information in business to those that focus more on the computer itself as a tool for the production of information. Nonetheless, all fifteen textbooks provide a hardware component. Typically, these textbooks describe the organization of the system as a whole, identify the fundamental components that make up the system, then discuss each component briefly. Each book provides at least minimal descriptions of the CPU, buses, memory, and common I/O devices, with simplified explanations of these devices and comparisons between different types of functionally similar devices. Most also address such hardware-related issues as the representation and manipulation of data in the computer, the layout, storage and retrieval of files, and the presentation of images and sound. Except for the discussion of the CPU, we concluded that the coverage of the system hardware as a whole is generally simple but adequate at an introductory level.

The single area in which we noted a significant deficiency is the description of the organization of the CPU. Fourteen of the fifteen textbooks present the organization of the CPU in terms of the traditional Control Unit/Arithmetic-Logic Unit model shown in figure 1. In this model the CPU is represented by an identifiable arithmetic-logic unit which performs logic and arithmetic operations using a set of registers within the ALU, and an identifiable control unit that controls the flow of instructions as they are fetched and executed. The words *Control Unit* and *Arithmetic-Logic Unit* are listed as keywords in nearly every text. Unfortunately, the CU/ALU model is no longer generally applicable to CPU organization, except perhaps in the abstract conceptual sense. Most modern computers do not contain specific components that can be labeled as the control unit or the arithmetic-logic unit.

The CU/ALU organization was first proposed by von Neumann for the EDVAC project in 1945. Most computer CPUs up through the mid-1980's were actually organized in this fashion. Older computer textbooks often featured photographs of CPU modules and micro-processors that identified the separate ALU and CU regions. Although the essential operations performed by the CPU have not changed, CPU organization has been significantly altered in the last ten years by advances in

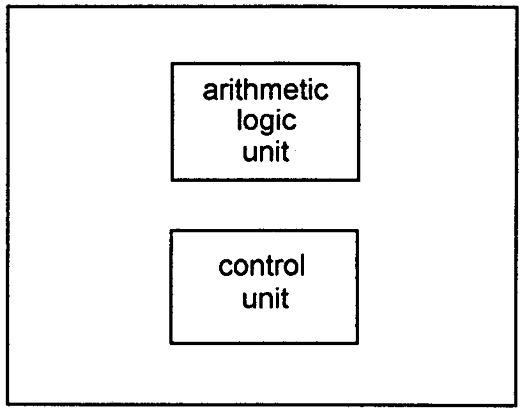


figure 1 -- Traditional CPU organizational model

technology, and by new processing techniques used to increase performance. In most modern computers, the "ALU" is only one element within an instruction execution unit, and the functions of the control unit are distributed throughout the CPU. Many modern CPU designs even provide multiple ALUs within a single CPU.

Alternative CPU Models

It is reasonable to assume that the primary goal for the use of a CPU model is to clarify the role of the CPU in the context of a computer system. To do so, an acceptable CPU model should meet three criteria:

- It should present a reasonably accurate representation of the CPU
- It should be simple
- It should support and enhance the student's understanding of CPU organization and operation

One approach for explaining the function of the CPU is to describe the tasks performed by the CPU without a model, or with an abstract model. Stallings [1996] describes the CPU in terms of five operations: *fetch instructions*, *interpret instructions*, *fetch data*, *process data*, and *write data*. More generally, the CPU fetches and executes instructions; these instructions move and manipulate data, and perform simple arithmetic and logical operations on data. For this purpose, the CPU can be described in terms of an instruction flow path and a data path. Hennessey and Patterson [1994] take a similar approach. They work from a classic abstract model that consists of five components, as shown in figure 2. The processor (CPU) is made up of the datapath and control components. In their view, "the *datapath* performs the arithmetic operation, and *control* tells the datapath, memory, and I/O devices what to do according to the wishes of the instructions of the program." This view is akin to the CU/ALU model, but differs in its acknowledgment of the distributed nature of each path.

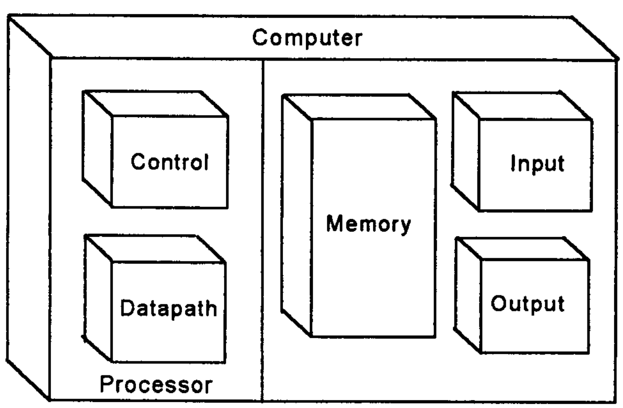


figure 2 -- Classic components of a computer

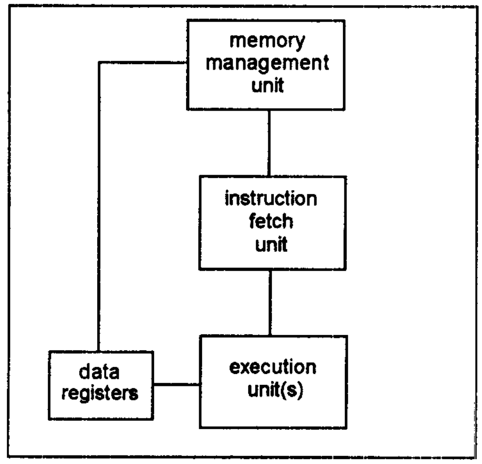


figure 3 -The instruction fetch/execution unit model for CPU organization

Figure 3 presents a simple alternative model that accurately represents the organization of most modern CPUs. Conceptually, instructions are fetched from memory one at a time and executed; the modern CPU is organized in such a way as to facilitate and enhance instruction processing. Instructions are fetched from memory, flow through each module, are processed, and are retired, preferably in an assembly line fashion.

The memory management unit provides a communication link between the CPU and the bus that provides access to memory for instructions and data. Its purpose is to fulfill requests for instructions and data, and to return data to memory. The memory management unit could be as simple as a bus interface unit that controls access to the memory bus. In most modern systems it also supports alternative forms of memory that may be present, in particular, cache memory and virtual memory.

Instructions are fetched by the instruction fetch unit. (Some vendors refer to this unit as the instruction unit or i-unit.) The instruction fetch unit also prepares instructions for execution by identifying the memory addresses and/or registers of the data required by the instruction. Most instruction fetch units provide a small amount of internal memory, so that several instructions may be fetched, decoded, and prepared before they are actually passed to an execution unit for execution. The Pentium Pro instruction unit can hold more than twenty instructions, for example, while they are being prepared for execution. It can also pass instructions to the execution unit out-of-order, bypassing instructions that are waiting for data in favor of later instructions that are ready for execution. In an abstract sense, the instruction fetch unit is analogous to the control unit of the earlier model, since much of the control for instruction flow is centered in this unit, however the instruction fetch unit provides many additional services, and control for the execution and memory management units is handled independently.

When the data required by an instruction is available, the instruction is passed to an execution unit for execution. An execution unit is functionally similar to the ALU. Arithmetic and logical calculations take place in the execution unit, in conjunction with one or more sets of general purpose registers. There may be several execution units operating in parallel. Most modern CPUs provide separate execution units for different types of data. In some CPUs, an execution unit is known as an e-unit.

Upon completion of execution, instructions are retired. In many systems, this requires no specific action, however, systems that permit out-of-order instruction execution and systems with multiple execution units must assure that instructions are completed in the sequence intended by the programmer. Some CPUs provide a separate retirement unit that writes data back to memory and registers in the correct instruction sequence. Other CPUs use special mechanisms within the instruction fetch unit to assure correct program execution. We did not include the retirement mechanism in our model because its role is not fundamental to CPU concepts, and does not meet our criteria for simplicity.

Benefits

The primary advantage to the use of the alternative model presented here is that it accurately represents the

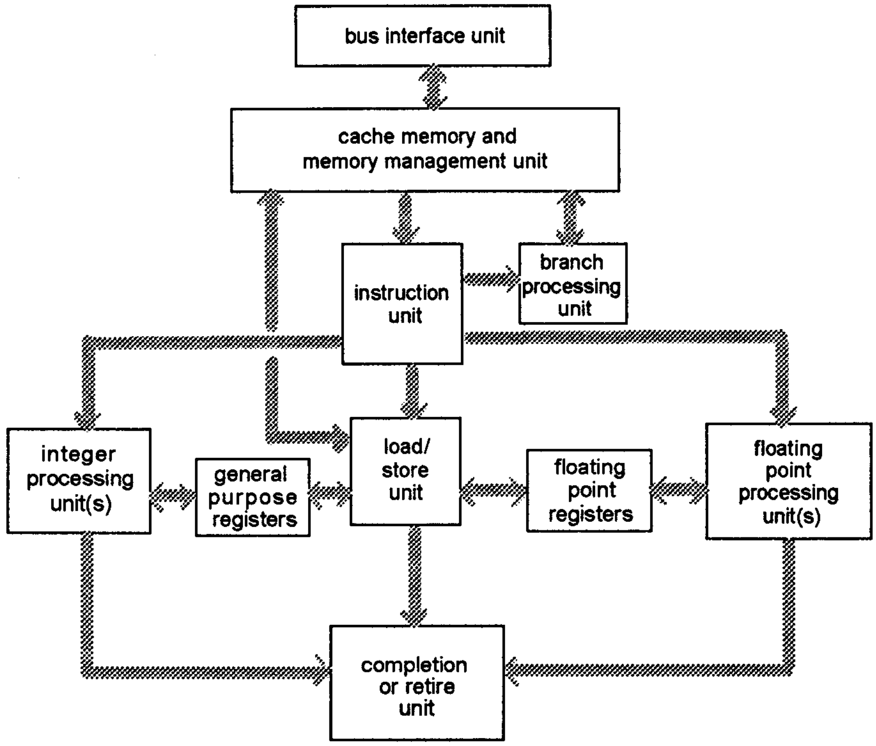


figure 4 -- PowerPC CPU block diagram

organization of a modern CPU. For comparison to a modern architecture, figure 4 shows the CPU block diagram for the PowerPC CPU. The Pentium Pro and IBM System/390 diagrams are very similar [Englander 1996]. An additional difficulty with the ALU/CU model is the lack of intuitive understanding that may be derived from each of the modules. The instruction fetch/execution unit model allows a straight-forward explanation of instruction flow and the instruction fetch-execute cycle that can be easily understood by students. My own experience indicates that the use of this model allows students to attain a clearer understanding of CPU organization and operation, with less effort.

Conclusion

The Instruction Fetch Unit/Execution Unit model meets the criteria as a suitable model for the description of CPU behavior. It is a simple, yet accurate representation of modern CPU organization. It improves student understanding of the role and operation of the CPU within a computer system. The author can also attest that it is easily extended to the more advanced concepts required for an IS'96/IS-4 compliant course in computer architecture.

References

Englander, I. S., **The Architecture of Computer Hardware and System Software: An Information Technology Approach**. New York: John Wiley & Sons, 1996.

Patterson, D. A. and J. L. Hennessey, **Computer Organization and Design, The Hardware/Software Interface**. San Francisco: Morgan Kaufmann 1994

Stallings, W., **Computer Organization and Architecture, 4th edition**. New York: Macmillan 1996

[the fifteen introductory textbooks surveyed are available as a list from the author]

Migrating to Windows NT: A Business Solution

J. Chris Hollowwa
Business Computer Systems
University of New Mexico
Albuquerque, New Mexico 87123
(505) 888-0354 jcholl@unm.edu

Abstract

The changing environment of computer operating systems and network operating systems has received a great deal of attention lately. This paper reviews the pertinent features of Windows NT, Microsoft's highly touted operating system. This paper is directed at business managers to inform them of the issues surrounding the implementation of Windows NT.

Introduction

Many business decision makers are facing tough choices regarding their information technology. Companies may be entrenched in the world of DOS, Windows, Novell, or a combination of the three. They may be using IBM's OS/2 or perhaps some flavor of Unix. To maintain a competitive advantage, companies must have effective computing resources and the ability to distribute those resources throughout the organization. Businesses which move to networked PC's need to decide on a platform and network operating system.

This paper looks at the history of Windows NT, its design goals, features, and compares it with other operating systems. I begin with the development of Windows NT, because background research is critical to any major business investment

Background: The Evolution of Windows NT

We begin our study of Windows NT by looking at earlier operating systems, and MS-DOS is a good starting point. DOS—with its sparse command line interface and an inability to do more than one operation at a time—left much to be desired. Macintosh and Amiga both had considerable advantages over DOS, and Unix was waiting patiently for its opportunity to become the preferred operating system.

In the late 1980s, Microsoft launched two new operating systems. Microsoft's solo venture was Windows, and it did not appear to be a winner at the outset. An IBM/Microsoft joint effort, termed OS/2, showed more promise. This operating system offered a true multitasking environment which could carry out several processes simultaneously and offer better memory protection. Unfortunately, Microsoft and IBM could not maintain their collaborative relationship and dissolved the partnership.

IBM took over OS/2 development but did not handle the marketing very well. Meanwhile, Microsoft was doing what it excels at: targeting the very users that IBM overlooked. Windows evolved gradually to the 3.1 version that would virtually sweep the marketplace. But Windows 3.x (3.1, and 3.11) still has limited capabilities and numerous annoyances.

It is the shortcomings of Windows 3.x that point toward its powerful, potential successor, Windows NT. Because Windows 3.x is based on top of MS-DOS, it is also tied to its limitations. With Windows NT, Microsoft has created a system that has the initial familiar appearance of Windows 3.x and. now, Windows 95. Underneath this shell is a true multitasking, robust operating system that many users have been demanding.

Design Parameters and Models

In the late 1980s, Microsoft began laying the groundwork for a new operating system, and hired David Cutler to initiate the Windows NT project. Cutler had been with Digital Equipment Corporation (DEC) and was no newcomer to the world of operating system development. He was responsible for a number of noteworthy operating systems, including VMS. He was to produce the operating system of the next decade, Windows New Technology (NT). Microsoft's Bill Gates met with planners and laid out definite market requirements and design goals for the new system. Key requirements included:

- *Multiprocessing capabilities*: The system should be capable of running on single and multiprocessor (more than one CPU) machines. On them very complex programs can be divided up between processors in order to share the computing load.

- *Networking capacity*: It was becoming obvious at the time that those using PC's would need to share files and peripherals. This would allow groups of inexpensive PCs to act much like a mainframe.
- *Secure environment*: Because of its network abilities, NT should offer extreme security for its resources. The U.S. government defined network security standards and Microsoft wanted to surpass them.
- *POSIX compliance*: The U.S. government requires operating systems under its contracts to meet POSIX standards. POSIX (Portable Operating System Interface based on Unix) calls for a common Unix-type interface so that programmers can port applications amongst other like systems (Custer, 1993).

In addition to the market requirements, the NT planners also had distinct design goals including:

- *Extensibility*: The designers wanted Windows NT to be able to grow and adapt easily over its life span.
- *Portability*: NT needed to be able to run on various processor platforms.
- *Reliability and Robustness*: These factors call for a system that performs predictably in the face of error conditions. For a system to be robust, it must provide safeguards for the operating system itself and for users' applications and files.
- *Compatibility*: Compatibility with other products in the Microsoft family as well as other file systems and network protocols was required.
- *Performance*: Performance is relative to compliance with the previously mentioned design goals. But planners still intended for Windows NT to run optimally on whatever platform it is implemented.

The NT development team began with Cutler and ten others. This number grew to about 40 or 50 people as the project progressed. Analysts estimate that, all told, Windows NT development involved over 200 people (Person, 1993).

The design of Windows NT was guided by three models in order to meet the lofty goals that had been set for it: 1) client/server, 2) multiprocessing, and 3) object.

Client/ Server

The client/server model refers to an approach where applications are distinctly separated from the operating system itself. Client/server methodology divides the operating system into a number of processes which implement *services*. These *servers* then respond to *clients'* requests for their services. Clients can be other parts of the operating system or an application program itself. A key advantage is that each operation is carried out in its own subsystem; if something goes awry, it does not effect other processes.Client/server also enables symmetric multiprocessing, a concept in the second NT model.

Multiprocessing

When an operating system is capable of managing multiple processes almost simultaneously, it is said to be capable of multitasking. Superior to multitasking is the ability to actually run multiple processes on a multiprocessor machine at the same time. Multiprocessing is seen in two forms, asymmetric and symmetric. In asymmetric multiprocessing, the operating systems usually need to be designed for the particular type of computer (proprietary). Since hardware design evolves constantly, it is not feasible to change an entire computing platform so regularly. NT developers used the symmetric multiprocessing model, which shares computing chores equally between CPUs, and is not dependent on proprietary technology. In this way, the power of multiple processors can be harnessed while still retaining a good degree of portability. Symmetric multiprocessing, whereby individual servers can work on different processes, is the preferred mode.

Object Model

Another model implemented in Windows NT is the object model. What we need to know about the object model design is that it allows NT to provide a secure, stable operating environment. It also tells us that NT is designed with a high degree of modularity. By working with flexible modules, Windows NT can be upgraded easily and ported to a variety of platforms such as Intel x86, DEC Alpha, PowerPC, or MIPS, without being completely rebuilt.

The key to Windows NT's ability to be ported to such a highly diverse group of processing environments lies in its use of the hardware abstraction layer (HAL). This layer lies between the operating system and the hardware itself. The HAL effectively hides the hardware-dependent aspects such as input/output interfaces and multiprocessor communication tools. In essence, the HAL becomes a translator between Windows NT and the various types of hardware platforms on which it runs. This allows for portability across a variety of computing hardware environments.

NT Basics: Requirements and Capabilities

We now move to the final product of Microsoft's Windows NT development (if there is such a thing). It is important to know not only what the hardware requirements of the operating system are, but also some of NT's basic capabilities. Microsoft lists the following basic system requirements for running Windows NT 3.51 on Intel-based systems: 386 or higher processor, 90MB of available hard-disk space, 12MB or more of RAM, and a 3.5 inch floppy drive (CD-ROM drive optional, but highly desired).

While the specifications state that Windows NT can run on a 386 computer with 12MB of RAM, these are low-end requirements for a Windows NT installation. In all likelihood, such a setup would barely run the operating system itself, let alone any applications. A review of Windows NT Workstation 3.51 by Mace and Miller in PC Magazine (September 26, 1995) yields a more realistic view. The authors state that the price of the stability and robustness provided by Windows NT is considerable system overhead. A system with a fast CPU and at least 16MB (preferably 32MB or more) of RAM was the bare minimum to adequately run Windows NT. The lesson is, if you want to implement NT, you need suitable hardware. If you are considering changing an entire division over to Windows NT, it is important that there is plenty of computing "horsepower" available.

Users involved with high-end applications will benefit the most from Windows NT. Packages grouped in the high-end category include: software development, financial analysis, scientific and business critical tasks, and engineering programs (such as CAD). Many of these types of programs have traditionally been run on mainframes and minicomputer workstations. With the ever-increasing power available in desktop microcomputers, much of this work can now be handled by far less expensive equipment.

Windows NT on Other Hardware Platforms

It is apparent that this operating system was designed to meet future computing demands. This orientation is particularly apparent when we look at leading edge technologies. These include many of the newest generation of CPUs—reduced instruction set computers (RISC) such as PowerPC or DEC Alpha, and CPUs using the Intel's x86 architecture employing complex instruction sets (CISC). The powerful new processors are designed to run best using the pure 32-bit code upon which Windows NT is based.

When programs are written using 32-bit code, many advantages can be realized: code is easier to write, memory protection and multitasking are enhanced, and large files are handled more efficiently. Yet there are disadvantages also. When a system is optimized for 32-bit programs, it is less efficient when a user runs 16-bit code. Programmers call it a "thunk" when the 32-bit operating system runs 16-bit code, and the ensuing conversion eats up processing time. Thus, current 16-bit programs might not run as fast with the newer processors compared to less sophisticated CPUs. In order to realize the benefits of Windows NT on new generation processors, 32-bit applications are needed.

Power PC and DEC Alpha

Much discussed in the computer world has been the advent of the RISC-based PowerPC architecture created by an IBM/Apple/Motorola consortium. IBM has been working on a version of OS/2 to run on its PowerPC systems but has faced multitudinous delays. With Microsoft's release of Windows NT version 3.51, there is now a very appropriate operating system for the PowerPC. A number of PowerPC servers are being marketed with Windows NT installed and IBM is one of the companies marketing such a package. The OS/2 for PowerPC operating system has been postponed indefinitely if not completely canceled by now.

Windows NT shows considerable potential on the PowerPC platform, but PowerPC is only one of many RISC processors. One of the hottest performing CPUs available is the DEC Alpha chip. With internal clock speeds greater than 300 MHz, DEC processors are a force to be reckoned with. Microsoft and Digital have collaborated, ensuring that DEC will show a strengthened commitment to the Windows NT operating system. In a *PC Week* article entitled "With DEC, Microsoft aims to rule beyond the desktop" (August 21, 1995), author Goldberg postulates that Microsoft has big plans for the DEC partnership. He estimates that Microsoft anticipates a time when it can meet customers' needs in the world of "Big Iron", just as it does in desktops. Goldberg believes that if Microsoft can control a group of customers beyond the desktop, then it will succeed in a manner that makes today's Microsoft look like a lemonade stand.

Complex Instruction Set (CISC) - x86

Windows NT appears to be an even more promising operating system for the Intel platform. The Pentium Pro (P6) has received negative press because it runs most current software (primarily16-bit) more slowly than its predecessor, the P5 Pentium. But the very reason that it underachieves with current software is what could make it and Windows NT the combination of choice into the 21st century. When the plans for the P6 were drawn up, it was assumed that we would be running 32-bit applications by now. That is not the case, and since the P6 was designed to wring the utmost out of 32-bit applications it falters running earlier versions.

Aside from being tuned for 32-bit code, the Pentium Pro was designed with multiprocessing in mind. The built-in ability to coexist with multiple processors (called "glue") is built right into the P6 architecture. Multiprocessor P6 PC's are being produced and the demand is already high. The P6 will be a mass-produced item and Intel will market it aggressively. With the rapid development of affordable, powerful processors, Windows NT's future looks bright.

Windows NT as a Network Operating System

NT is also a full-featured network operating system. This section will focus on the Server version of Windows NT which is designed for network connectivity and management.

An initial concern might be that Windows NT would only support Microsoft based network setups. However, Windows NT Server accommodates Novell NetWare, AppleTalk, TCP/IP, and numerous other options. Microsoft realizes that today's workplace can offer a myriad of computing network platforms that need to be tied together.

Microsoft states that Windows NT Server combines the file and print services of NetWare with the application capabilities of Unix in one multipurpose network operating system. As a file and print server, Windows NT Server allows you to share information and access printers and other devices across the network. Moving to Windows NT Server doesn't mean giving up your current systems. It is designed to integrate and unify a network. For example, you can add file servers and mission-critical application servers into your NetWare network, deploy file and print servers to Unix environment, and bring mainframe sales data directly to Microsoft Excel users.

A number of articles in the trade journals and computer magazines support NT Server's claims with actual performance. Consider first the competition between Windows NT and Unix.

Windows NT Compared with Unix

Unix has been an industrial-strength operating system (and network operating system) of choice since its inception in the early 1970s. There are many different versions (or flavors) that can run on just about any type of hardware. Unix offered freedom from legacy computer systems (IBM, DEC) and offered information systems shops an "open system" where less expensive hardware could be mixed and matched. However, the very fact that Unix comes in so many flavors has also been its major detraction. So many developers have had a part in the evolution of Unix that to date, it has been impossible to come up with a single, unified version.

Windows NT is a new operating system compared to Unix, and business has been reluctant to port its mission-critical operations to NT. A critical review of Windows NT was presented by Carl Dichter (Unix Review, May, 1995) where he stated that the Windows NT 3.5 was Microsoft's first workable model of a true 32-bit operating system. NT 3.5 was commended for being able to run faster with less memory than its predecessor. Windows NT's hardware abstraction layer was also praised for its role in application portability.

Dichter did criticize NT on a number of fronts, especially for hardware and software incompatibilities and inadequate POSIX compliance. NT was cited for its lack of support for many DOS commands, and the unavailability of NT applications. The author would not recommend NT because of its high demand for RAM and its incompatibility with existing desktop applications. Dichter predicted slow growth for the operating system but acknowledged it had some potential as a network server.

Not long after this article was published, it became fairly common to see PC's available with 16MB of RAM and a Pentium 133 or faster processor for less than $2000. While corporations might not be able to purchase quantities of new, relatively inexpensive powerhouses now, they will be buying them soon. Organizations that deal with the types of applications for which NT is optimized (finance, engineering, science) will be able to (and need to) purchase computers able to run NT well.

The battle lines have been drawn between Windows NT and Unix. Many diehard Unix proponents cannot believe that Microsoft has the gall to think that its NT operating system can compete with Unix, much less displace it. An article by Gurton in *Computer Weekly* (April 20, 1995) relayed the proceedings of the Uniforum '95 trade show. Normally a Unix stronghold, Uniforum has been invaded by Windows NT and over 80 percent of the 1995 exhibitors had NT products or supported NT compatibility with their Unix systems.

Silicon Graphics CEO Ed McCracken was particularly vocal in denouncing Windows NT. "I'd like to go on record and say that Silicon Graphics has not supported NT, is not supporting NT and has no development under way of any kind to support NT. ...Unix has been, and remains, the operating system of choice..."

Microsoft volleyed back stating that "...companies such as Silicon Graphics and others who defend Unix so fiercely have a deep fear of the PC market-place and cannot accept the shift in the power/price ratio on the desktop. ...it is like the old days when the mainframe community resisted the minicomputers."

Despite the ongoing debate, it is incontrovertible that Windows NT is increasingly popular and could be the harbinger of an exodus from the minicomputer. Numerous articles report the NT challenge to Unix and the consensus seems to be that Windows NT is not as scaleable as Unix. However, Unix is present in so many versions that it has not really elicited the creation of prepackaged software, which is Microsoft's forte. If businesses can purchase inexpensive, off-the-shelf applications that will run on a variety of platforms, users will probably flock to them.

Windows NT vs. Novell Netware

Not only has Windows NT become an option to Unix, it is hoping to challenge the king of pure networking: Novell Netware. Novell claims to be the Network Operating System (NOS) of choice on about 60 to 70 percent of all PC-based networks. According to an article in the LAN Times (August, 1995), author Kaplan states that NetWare's dominance will drop to around 50percent of PC networks by 1997. Microsoft is projected to move from its current 9 percent to a 15percent market share in 1997.

The LAN Times article actually paints a rather grim picture for NT's bid for the corporate network. It stated that Netware performs better on the network and has better file and print services. It was also noted that third party software support is better for Netware, and Novell's own customer support is superior to Microsoft's.

While most current comparisons of NetWare and NT state that NetWare is the superior file and print server NOS, others give NT a little more credit. In a *Communications Week* article, "The great NOS shoot-out" (June 19, 1995), Mier and Mier compared IBM LAN Server 4.0, Windows NT 3.51, and Novell NetWare 4.1. In the "ease of use" category, NT and NetWare ranked equally well, despite the fact that they function quite differently. NT Server was lauded for excellent on-line help and documentation as well as the best client-to-server interface. NetWare excelled in the ease of use of the administrator's tools. Tasks like creating a new user were far simpler with NetWare tools than with those packaged with Windows NT Server.

NT Server was ranked highest in the management and administration category and was lauded for its ability to centrally manage multiple servers. Many of its utilities were rated as superb. But as for actual performance, the *Communications Week* article concurred with most other articles on this point: Novell Netware is the top *performing* NOS. The authors did state, however, that the margin was not that considerable. The final decision of the NOS competition was that the race between Windows NT Server and NetWare 4.1 was indeed a close one. Based on the magazine's grading criteria and scoring methodology, Microsoft was awarded a clear win.

In the NetWatch column in the *Computer Shopper* (October, 1995), author Rigney also admitted that the NetWare vs. NT Server comparison revealed no real winner. But, by virtue of this, he gave the advantage to Novell because it is already the network standard and it will be tough to unseat NetWare. Rigney and other computer columnists have praised Windows NT for its ability to work well in conjunction with NetWare. All in all, Windows NT Server gets good marks from many thorough comparisons between it and Novell NetWare.

Conclusion

The problem with writing an article on a topic like Windows NT is that there is always something more to be said about the subject. While attempting to wrap up my ideas regarding the key points of the NT operating and networking system, more and more information appeared. Here are a few recent developments on the NT scene.

- The Windows 95 Shell has been successfully grafted on to Windows NT in version 4.0 along with a number of other useful features. Windows NT 4.0 was officially released to systems manufacturers on July 31, 1996, ahead of schedule.
- Windows NT Web server products for the Internet are becoming popular. Internet Information Server and Front Page are but a few of the tools Microsoft now bundles with the server version of NT 4.0.
- Microsoft Cairo, code name for the next generation of Windows NT, promises to be a true object-oriented version of NT. A beta version is supposed to become available some time in 1997.
- Clustering—a computing environment created by connecting servers via a shared SCSI bus—ensures system availability in the event of a failure. To end user clients, a cluster appears as a single system. DEC developed this concept over ten years ago and it has evolved as a joint venture between DEC and Microsoft.

The topics of Windows NT, its Internet and Intranet capabilities, and its position against Novell Netware are popular in the computer press. It is difficult to pick up a trade magazine and not find something pertaining to the subject on the front page. How much of this is media hype? We have acknowledged that Microsoft excels in marketing and generating publicity. Needless to say, there is no shortage of discussion about Windows NT and Microsoft's plans for it.

It has not been the intent of this paper to be an ardent Microsoft cheerleader. Whenever possible, the criticisms of the operating system have been noted as well as the strengths. But from an objective point of view, Windows NT appears to be a viable product. With its capability to work on various multiprocessor platforms, built-in networking, and a robust nature, Windows NT has significant potential. While it may not be the ideal operating system for all workplaces, it has the features that could make it a good fit for many "power user" and networked environments.

Whether or not Windows NT can unseat industry standards like Novell Netware and Unix is hard to determine. Will the upcoming versions deliver on all of Microsoft's grand promises? These questions can only be answered after considerable experience and testing. In the meantime, it would behoove anyone with information system responsibilities to take a hard look at the evolving race between computer platforms. Windows NT can run on relatively inexpensive computer architectures such as the Intel Pentium Pro and can provide a secure, stable, and very robust platform for applications traditionally run in Unix or Novell environments. All in all, Windows NT warrants serious consideration as a mission-critical operating system.

Acknowledgments

The author would like to thank Professor Laurie Schatzberg, Lisa Hogan, and Shirley Turner for their skillful help in editing this article. Also, thanks to Larry Desonier and Sandia National Laboratories for providing excellent advice and plentiful resources.

References

Custer, H. (1993). Inside Windows NT. Washington: Microsoft Press

Dichter, C. & Bailey, D. (1995, May). Windows NT 3.5. Unix Review, pp.73 (4).

Fawcett, N. & Vowler, J. (1995, January 12). Will NT be king? Computer Weekly, pp. 30 (2).

Goldberg, A. (1995, August 21). With DEC, Microsoft aims to rule beyond the desktop. PC Week, p. A12 (1).

Gurton, A. (1995, April 20). No punches pulled: Unix vs. NT. Computer Weekly, p. 44 (1).

Kaplan, S. (1995, August 14). Windows NT reality check: the Microsoft machine at work. LAN Times, p. 64 (1).

Kay, A.S. (1995, April 24). Windows NT cleans up: Microsoft is sweeping up sales among Unix users wanting a more unified NOS. Communications Week, 39 (4).

Mace, T. & Miller, M.J. (1995, September, 26). Architectural review (differences between IBM OS/2 Warp, Windows 95, Windows NT and Windows 3.1) (Your Next OS). PC Magazine, pp. 132 (6).

Microsoft [Online] (1995). Windows NT references. Available: http://www.microsoft.com/

Mier, E.E., Mier, D.C., & Smithers, R.J. (1995, June 19). The NOS shoot-out. Communications Week, p. 46 (5).

Person, R. et al (1993). Using Windows NT. Indianapolis: Que Corporation.

Rigney, S. (October, 1995). Windows NT looks better all the time. Computer Shopper, p. 542 (2).

Suggested Readings

Chacon, M. & King, C. (1995, June). NT: older and wiser. LAN Magazine, pp.51 (4).

Chernicoff, D., Jensen, B. & Salvator, D. (1995, October). The great debate: NetWare 4.1 vs. NT 3.51. Windows Sources, pp. 76 (2).

Clark, D. (1995, August 24). Microsoft Windows 95 is just a sideshow; company's growth hinges on NT OS for corporate networks. The Wall Street Journal, p.B8(E) col. 1.

Davis, F. (1995, December). Windows NT Server has found its killer app: the Internet. Windows Magazine, pp. 59 (2).

Delson, D. (1995, August 21). Whatever you do, don't migrate until you have given NT a long, hard look. Info World, p. 66 (2).

iNformaTion, [Online] (1995). Windows NT FAQ's, websites, etc. Available: http://www.rmm.com.

Lipschutz, R.P. (1995, May 30). Microsoft Windows NT Server 3.5. PC Magazine, pp. 238 (4).

Pontin, J. (1995, September 11). Road to fork for Win95, NT: OSes will share technologies but retain separate code bases. Info World, pp. 1 (2).

Reichard, K. (1995, October 10). Windows NT-based servers: what's the rush? PC Magazine, pp. 229 (6).

A Typology for Distance Learning: Moving from a Batch to an On-line Educational Delivery System

Submitted to the Information Systems Education Conference (ISECON):

Caroline Howard
Richard Discenza
College of Business
University of Colorado, Colorado Springs
P.O. Box 7150
1420 Austin Bluffs Parkway
Colorado Springs, CO 80933-7150

A Typology for Distance Learning: Moving from a Batch to an On-line Educational Delivery System

Abstract

Distance education offers some major advantages in this type of environment. This paper develops an extension of the DeSanctis and Gallupe (1985) typology for groupware systems by incorporating a third dimension, communication. Distance learning is defined as we know it today, and categories for delivery systems are specified. The quality parameters for distance learning are analyzed for student assessment and program evaluation.

Introduction

Today's post-secondary institutions of business education face major challenges as we near the 21st century. While the number of institutions providing business degrees has increased dramatically since the 1970's, the number of traditional students attending business school immediately after high school is decreasing forcing business schools to compete for a finite group of students (Kahn et. al 1995). At the same time the survival of institutions of higher education is increasingly dependent on their ability to provide increasing levels of service to their constituents with decreasing revenues (Kahn et al. 1995).

In recent years distance education programs have proliferated to meet these changing needs by allowing educational institutions to provide quality education to an expanded population of students. Distance education, or the delivery of classes to geographically-dispersed students, who are unable to attend formal classroom settings to take classes and pursue degree or certificate programs. For the past twenty-five years the Open University in Great Britain has granted degrees to over 100,000 students in accordance with its mission of making higher education accessible to all sectors of society (Romiszowski 1992).

In the Western United States where many population groups are located too far away from a campus to allow class attendance, 150 colleges and universities have formed the Western Cooperative for Educational Telecommunications Consortium to provide distance education to these populations. In the International arena Mexico, China, Canada and the United Kingdom provide televised classes and computer conferencing to their rural or disabled students who are unable to travel to campus to attend classes.

Distance education allows colleges and universities to better meet the needs of non-traditional students. This is particularly important as the proportion of these students is increasing while the proportion of traditional students who attend college on a full-time basis immediately after high school is decreasing (Appellate 1993). These students are likely to be working adults, many with family demands, who are returning to college to obtain a degree, receive education necessary to change careers or continue the lifelong education necessary to succeed in our society (Holloway and Ohler 1993). While traditional educational approaches may suffice for younger and senior-aged students, where traveling to campus for a quarter, semester, or longer period of time is possible, these programs do not adequately address the needs of non-traditional, career-oriented students.

Distance education is especially attractive to institutions of higher education, because it is both cost effective and has been shown to maintain the high educational standards to which colleges and universities aspire. Technological advances accompanied by decreasing costs have made these programs increasingly cost effective. Because of the opportunities for expanding student populations while maintaining quality education, distance education programs are proliferating. Thirty-seven states currently provide distance education through their public education networks (Ely 1991), and degree programs exist in universities and colleges throughout Europe, Asia, the United States, Canada and the Far East (Ely 1991). Finally, studies evaluating the outcomes of distance education have consistently shown positive results and no differences in terms of achievement between distance and traditional education (Cheng et al. 1991).

This paper will explore the many forms of distance education through the development of a typology which categorizes distance education along three dimensions: 1. direction of communication flow, 2. predictability of physical location and 3. communication technology used. The typology substantially modifies the groupware typology developed by DeSanctis and Gallupe (1985) so that it better applies to distance education.

Initially, the dimensions of the typology are defined. Next, the various types of distance education are described in terms of the typology's dimensions. Finally, other aspects of distance education not covered in the typology are explored.

Typology for Distance Education

The typology describes distance education using the three dimensions of technology, flow and location which largely determine the nature of the educational experience. This section will describe the nature of each of these dimensions and then the typology will be used to categorize the types of distance education that are currently in use.

Flow. The direction of the communication flow determines the interaction students can have with the instructor and with each other. Three types of communication flow are defined in the typology as shown in Figure 1. *Simplex communication* is only from the instructor location to the student location but not from student location to that of the instructor. This type of flow does not allow the student to respond to the instructor or to provide feedback in a timely manner. Typically, the student vehicle for responses and feedback in these courses is the United States mail.

INSERT FIGURE 1 ABOUT HERE

Half-Duplex communication is transmission in both directions, from student to instructor and instructor to student. However, communications can only occur in one direction at a time. Usually, instructors broadcast their courses and later students provide feedback or comments. Sometimes the communication media used for student responses is not the same as the technology used for instructor broadcast. Students often use computer conferencing or electronic mail for their feedback and responses.

Full Duplex communication allow simultaneous communication in both directions. Therefore, courses can be truly interactive with both the instructor and students benefiting from real-time interactions, questions and other responses. Due to technological advances and reductions in technology costs, the future will bring more and more on-line courses using Full Duplex communications.

Location. Students can be located in the same location as the instructor, different but predictable locations or different and unpredictable locations. These are shown in Figure 1. Obviously, distance education only exists when students and instructors are located in different places. Whether or not students are located in *predictable* or *unpredictable* locations is important to consider, because predetermined student locations can be equipped with expensive receiving and transmission equipment such as sophisticated computer conferencing systems or interactive television. This specialized equipment is too costly for the individual student. Also, students located individually do not have the face to face interaction with other students that students located together have.

Technology. The type of technology used for course transmission determines the types of instruction which can be used. Referring to Figure 1, courses can be transmitted via *audio* (audio tape, telephone radio), *video* (television or video conferencing), computer conferencing (may also use video or audio transmissions) or mixed media (a combination of two or more approaches). At the present time *video* television transmission is the richest form of communication used to transmit video and audio information to individual students and is frequently located in their homes (Daft and Lengel). To transmit courses to groups of students sophisticated computer conferencing systems, incorporating both video and audio conferencing located in business conference settings or sophisticated classrooms, are sometimes used (Dutra 1996).

As the costs of computer technologies decrease, more individuals will have access to more sophisticated computer conferencing systems in their homes. In the meantime many distance courses use television for transmission to students while using electronic mail for two way communication between instructor and student. The typology can be a very useful tool for categorizing the various forms of distance education in use. Using Figure 1 as a basis, the following examples fit nicely into the matrix.

Operationalization of the Typology

The typology divides location into *predictable* and *unpredictable* locations. As the examples in this section illustrate, whether or not location is predictable has important implications for distance education. Usually, student location is predictable when students are grouped in specially equipped classrooms or library locations. These locations can be equipped with specialized communication technology. Since these technologies are often very expensive, it is not feasible to rely on the existence of specialized technologies when students are in unpredictable locations, usually their homes.

For example, schools can use their intercom systems for the broadcasting of speeches, music or other messages throughout the school (Figure 2: 1) or have computer assisted learning software available for use in their libraries or classrooms on specially equipped multimedia systems (Figure 2: 7). Sophisticated multimedia systems can combine computerized training with audio or video transmissions for use in the classroom or individually (Figure 2: 10).

INSERT FIGURE 2 ABOUT HERE

Some types of distance education are similar for predictable and unpredictable locations. Computerized training is an example. However, many students only have assess to the hardware and software in their libraries or schools.

Although video course transmission is replacing audio to a large extent, audio teleconferencing is still used by universities and secondary schools to broadcast courses to predefined locations (Hugdahl 1982). When there are a large number of listening locations, it is impossible to maintain the on-line interaction with all locations simultaneously possible using teleconferencing (Figure 2: 3). In these situations audio interactions occur in only one direction at a time (Figure 2: 2). An example of this type of course is the University of Wisconsin's broadcasting music courses to 200 listening locations in public libraries, hospitals, other civic locations, organizations and the University's twenty-six campus locations (Hugdahl 1988).

Secondary schools and institutions of higher education are taking advantage of the potential and affordability of video communications for distance education. Schools at all levels are investing in video classrooms to receive classes broadcast from a central location (Figure 2: 5). Twenty-five states broadcast televised classes and have recipient students in specially equipped classrooms (Barker 1987). These states give their secondary and college age students a wider range of courses to choose from by broadcasting courses from a central location to individual classroom settings located throughout their states. As part of a statewide initiative promoting distance education, secondary schools in Utah use televised courses to augment their course offerings while avoiding the hiring of additional personnel to develop and teach elective courses.

In addition, colleges and universities, government and military organizations including the U.S. Navy and Congress, as well as numerous business organizations have invested in interactive television systems for coursework and training (Figure 2: 6). A number of colleges, including Minot State College, Stanford, and California State(Chico), have used interactive television to successfully conduct courses.

Often, there is more than one technology used for broadcasting distance education. Television programs broadcast to schools are sometimes accompanied by computer assisted learning materials which allow the student to further pursue the topic at his or her own pace (Figure 2: 4). Frequently, due to the expense of interactive television, students respond to televised broadcasts using audio or electronic responses (Figure 2: 11). On the other hand, as the costs of technologies decrease, business and some educational organizations may use video and computer conferencing systems to transmit training or course information to distributed groups of employees located in rooms equipped with the appropriate hardware (Figure 2: 12). For example, SUN and Apple use computers equipped with video cameras for their interactive distance training.

When students are not in predictable locations, courses usually rely on more prevalent and affordable technologies for course reception which students are likely to have in their homes. Early forms of distance education relied on one way or simplex communication from the instructor to the student. Instructors frequently used audio or video transmissions to communicate to the student while the U.S. mail was used for distribution of coursework, submission of assignments and all other correspondence. Due to the length of time from sending to receipt of material using the U.S. mail, this form of communication is considered simplex in this typology.

The use of radio and audiotapes for communicating course content to unpredictable individual student locations has been a cost effective, easy to administer form of distance education for decades (Figure 3: 1). Most students have had easy access to radios and tape players. Therefore, correspondence schools have traditionally included audiotapes in their course materials. However, since most individuals own televisions and video cassette recorders, audio courses have largely been replaced by video.

INSERT FIGURE 3 ABOUT HERE

The most common form of video courses with one direction transmission is the telecourse (Figure 3:2). Hundreds of colleges and universities in thirty -seven states currently use Public Television to broadcast their courses. These are a viable way for non-traditional, remote and handicapped students to easily benefit from college courses which they are unable to physically attend. (Catchploe1991). The Western Cooperative of 150 colleges and universities throughout Alaska, Arizona, California, Colorado, Hawaii, Idaho, Montana, Nevada, New Mexico, North Dakota, Oregon, South Dakota, Utah, Washington and Wyoming uses telecourses to provide college coursework to rural, handicapped and special need students.

These courses offer student flexibility in both the 74time and place in which they attend class. Students can view the programs from any television receiving the broadcast station or can tape the classes for later viewing. Taping the courses also allows students to review sections of the course with which they are having difficulty or to rapidly advance through sections which are repetitive or easy.

Business and educational organizations (Kahn et al. 1995) are increasingly using electronic mail, the Internet or the World Wide Web to transmit informational materials to students (Figure 3:7). These courses may use electronic mail over the Internet (Figure 3:8) or have on-line discussions between groups (Figure 3:9) using electronic mail or groupware.

Many of these courses will send information using more than one technology. Students may supplement a televised class with a computer tutorial. Since students can not usually afford interactive television or video conferencing, course designers will often have students communicate using electronic mail (Figure 3:11) or participate in computerized group discussions (Figure

3:12). Usually, student interaction is conducted using an alternative method of communication.

Implications for Further Research

As can be seen in previous sections, the typology can be used to classify distance education. In future research, other dimensions of distance education are important to consider in the research. The social interactions of students are dependent on the number of students taking a class together. For individual students viewing distance courses from their homes, they may not receive the educational benefits which result from interacting with their peers. Also, these students may be lonely. The United Kingdom encourages computerized student discussions and suggests that students visit the campus for an orientation during the degree program.

Class size is a determinant of the type and quality of interaction. When classes become too big, it is impossible to have students interact with each other. The United Kingdom's Open University uses computer and video conferencing to transmit courses to thousands of students. They call these courses "stadium classes". Obviously, not all the students can interact with each other. To allow students to have quality interactions, they have computerized "breakout rooms" where students can have computerized discussions with a limited number of students. Finally, the use of a facilitator has been found to be critical to the success of these classes.

Figure 1: A Distance Learning Typology with Space, Time and Communication

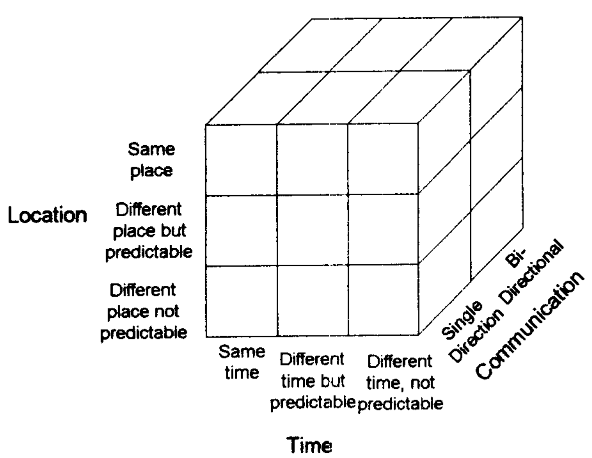


Figure 2: Operationalization of the Distance Learning Typology for Predictable Locations

Communication Technology	Simplex: *One direction only*	Half-Duplex: *One direction at a time*	Full Duplex: *Both directions simultaneously*
Audio	Intercom transmissions 1	Audio broadcasts to classrooms equipped to listen & provide subsequent response using microphone 2	Teleconferencing to classroom 3
Video	Televised courses to specially equipped classrooms 4	Televised courses to classrooms equipped with equipment for subsequent video response 5	Interactive television and video conferencing in equipped classroom 6
Computer	Educational software distributed to libraries and electronic classrooms 7	Course materials distributed on the Internet, World Wide Web or electronic mail to classroom or library 8	Computer conferencing in equipped classroom 9
Mixed	Computerized training with audio or video transmission to classroom 10	Televised courses with electronic response or audio by microphone 11	Video and computer conferencing combined 12

Direction of Communication Flow

Figure 3: Operationalization of the Distance Learning Typology for Unpredictable Locations

Communication Technology	Simplex: *One direction only*	Half-Duplex: *One direction at a time*	Full Duplex: *Both directions simultaneously*
Audio	Radio transmitted or audio-taped courses 1	Radio or audio-taped course with telephone response 2	B Teleconferencing 3
Video	Televised courses on public networks 4	F Televised courses with subsequent student video response 5	B F Video-conferencing and picture phones 6
Computer	B Computer assisted learning or tutorials 7	B Course materials distributed on the Internet, World Wide Web or electronic mail 8	B Courses conducted using groupware or computer discussions 9
Mixed	B Televised courses with individual computerized learning component 10	Televised courses with electronic mail or Internet response 11	Television and computer discussions combined 12

Direction of Communication Flow

References

Albrektson, J. R. (1995) "Mentored Online Seminar: A Model for Graduate-Level Distance Learning" The Journal of Technical Horizons in Education 23(3) pp. 102-105.

Birchalll, D.W. and L.A. Wilson "Interaction Versus Independence: The Dilemmas in Designing a Computer Mediated Communications Student Support System in A. Tait (ed), Interaction and indepdence: Student support in distance education and open learning: Conferenc Papers, Cambridge: international Council for Distance Education and the British Open University Regional Academic Services, 25-32.

Gao, F. W.Li and D.Li "A PC-based Audio/Graphics/ Video Image Stellite Communicatin Education Systems and its Applications. Educational and Training Technology International,.26(3) 248-53.

Andriole, S.J., Lytle, R.H. and Monsanto, C.A. "Asynchronous Learning Networks: Drexel's Experience", The Journal of Technical Horizons in Education 23(3) 97-102.

Applegate, L. (1993) Leveraging Diversity A presentation to the International Conference on Information Systems, Orlando, Florida.

Barker, B.O. (1987), Interactive learning by satellite, The Clearinghouse, 61(1) 13-16.

Brock, D. (1990) "Using Technology to Deliver Education" Bulletin of the American Society for Information Science 16(6) 9-11.

Catchploe, M.J. "Student Response to a Distance education course incorporating live, interactive television" The American Jorunal of Distance Education, 3(2),23-41

Cheng, H.C., Lehman, J. and Armstrong, P. (1991), Comparision of performance and attitude in traditional and computer conferencing classes", The American Jorunal of Distance Education, 5(3),51-64.

Cookson, P.S. (1989) "Research on Learners and Learning" The American Journal for Distance Education" The American Journal of Distance Education, 3(2) 22-34.

DeSanctis, G., & B. Gallupe, (1985) "Group Decision Support Systems:A New Frontier", DATA BASE 3-10.

Ely, Donald P.(1991) "Trends and Issues in Educational Technology, 1989" in Anglin, G. (ed.) Instructional Technology Englewood, CO:Libraries Unlimited, Inc. 34-58.

Grudin, J. (1994) "Computer Supported Cooperative Work: History and Focus" Computer 27(5) 19-26.

Holloway, Robert E. and Ohler, Jason, (1993) "Distance Education in the Next Decade" in Instructional Technology Englewood, CO:Libraries Unlimited, Inc. 250-275.

Hugdahl, E.O. (1982) "Once you've got a communications system, how to keep people using it or keeping the show on the road" In Teleconferencing and Electronic Communications: Applications, Technologies, and Human Factors comp. L.A. Parker and C.H. Olgren, 253-268, Madison:Center for Interactive Programs, University of Wisconsin-Extension.

Hugdahl, E.O. "Meeting Needs of Church Musicians by Audio Teleconferencing in Wisconsin" The American Jorunal of Distance Education, 5 (4),63-70

Kahn, B.K. "The challenge of IS Education in Business Schools: Coping with ever-changing Demands of Students' Diversity, Business Expectations and Accrediation", Panel Discusion at the International Information Systems Conference.

Romiszowski, A.J. (1992) Computer mediated communication: A selected bibliography. Englewood Cliffs, NJ: Educational Technology Publications.

Shepherd, M. (1995) "An Empirical Analysis of the Factors Affecting the Diffusion of Group Support Systems in Organizations" Doctoral Dissertation, Graduate School of Management, University of Arizona.

Wagner, E.D. and B.L. McCombs, (1995) "Learner Centered Psychological Principles in Practice: Designs for Distance Education". Educational Technology 25(2) pp, 32-35.

Wolcott, L. (1995) "The Distance Teacher as Reflective Practitioner", Educational Technology, 25(1) 39-43.

Quantitative Tools: Do They Really Matter in TQM?

Larry Scheuermann, DBA
Hal Langford, Ph.D.
Sandra B. Scheuermann, MBA, CPA
University of Southwestern Louisiana
PO Box 43570
Lafayette, Louisiana 70504
les2097@usl.edu

Introduction

Total quality management (TQM) can play an important role in helping U.S. companies recover their technological proficiency. Companies such as Xerox and Motorola have emerged as leaders in their industries as the result of implementing TQM [1]. TQM can be defined as an approach to planning and implementing continuous organizational improvement. Its focus is on satisfying customers' desires, identifying problems, building commitment, and encouraging open decision making among workers [18].

TQM involves two major components. The first component is the use of statistical tools such as flowcharting, Pareto analysis, cause-and-effect diagrams, and statistical process control in the decision-making process. These tools help abolish nonvalue-added activities in all functions [1]. The second component is a change in managers' behavior: going from managers who direct, are competitive, rely on rules and organizational hierarchy, make separate decisions, view people as costs, and promote sameness, privacy, and passivity to managers who lead, guide, cooperate, focus on the process, use informal networks, view people as assets, and encourage variety, flexibility, openness, sharing, risk taking, and involvement [1].

Both of TQM's elements are implemented with the intention of meeting customers' needs. TQM's effective implementation relies on developing a business plan established on core skills and providing value-added, customer-focused processes, whether it be in a hospital or a service type industry [1].

This research examines the association of several variables that are used in TQM implementation in hospitals, manufacturing, and service organizations. The variables are analyzed in two groupings: qualitative measures of TQM and quantitative measures of TQM. The study depicts the successfulness of TQM and whether qualitative or quantitative measures are a determinant of this success.

Beginning with a brief background on supporting literature of this type of research the study then presents the hypothesis to be tested. The research design and analysis techniques are discussed next. Finally, the results and conclusions are presented.

Literature Review

A search of available literature did not reveal any studies testing the effectiveness of Statistical Process Control to the success of TQM programs. However, many articles have been written concerning the factors for a successful TQM program. Several authors have stated that for TQM programs to be successful in any organization, factors such as management commitment (including the CEO), employee involvement (teamwork), cultural change, ongoing training, communication, and commitment to quality must be implemented into the organization [2] [3] [4] [5] [6] [7] [8] [9] [10] [12] [13] [14] [15] [16].

Many U.S. organizations, such as L.L. Bean Inc., General Motors Corp., Universal Foods Corp., and Hewlett Packard Co., are actively dealing with the issues of being on the mature and declining part of the success cycle as a result of losing market share to foreign competitors. These organizations are implementing TQM and are achieving success by implementing it correctly in their organization, while at the same time keeping employees involved in this ongoing process [2] [3] [10] [17].

TQM involves the combination of both qualitative and quantitative techniques. Companies thus far have placed more emphasis on the qualitative aspects of TQM; however, companies will not realize the full advantages of TQM until quantitative techniques are integrated into their operations [11].

Research Hypothesis

TQM programs are not always effective. This research attempts to identify those factors contributing to success/failure of TQM efforts. The research hypothesis is: ***TQM programs are successful in organizations when their approach to TQM is based on the application of Statistical Process Control tools/principles rather than solely on qualitative measures.***

Methodology

A questionnaire was developed and mailed to 1,484 Quality Managers in a four-state area (Louisiana, Texas, Arkansas, and Mississippi). The questions addressed the following issues:

Classification of organizations
Degree of success with TQM
Determination of existing TQM program
Use of quantitative and qualitative tools
Years to implement TQM program
TQM framework
Results of customer research
Initial training time
Improvement in quality of product
Abandonment of TQM.

A pilot study was done to test the questionnaire's adequacy. The pilot study was conducted at University Medical Center in Lafayette, Louisiana. A total of five individuals participated in this test. Modifications to the questionnaire were made after this was completed.

Descriptive statistics, Spearman-rank correlation, and logistic regression were used for the analysis performed in this study. Descriptive statistics were used to formulate initial ideas of the data that were then compared to the statistical analysis to form the conclusion of the study. The Spearman-rank correlation statistic was used to measure the association between two variables (quantitative and qualitative tools). Due to the discrete nature of the variables, logistic regression was performed to determine whether the model explains a significant amount of variation of the dependent variable.

Results

Descriptive Statistics

The response rate of the survey totaled 365 returned surveys, or 25%. Table 1 show the breakdown in frequency and percentage of the usable responses received from the mail-out of the questionnaire.

Of the returned questionnaires, 34, or 9%, have not implemented TQM. These responses were omitted from the sample. Manufacturing represents, by far, the largest number of responses at 184, or 56%, hospitals at 92, or 28%, and service organizations at 55, or 17%. Implementation of TQM, as shown in Table 2, is at high levels, suggesting that TQM is very popular at this time. Hospitals have a very high implementation rate due to new accreditation standards in 1994. Because of these new standards, many hospitals are new to some aspects of TQM. This is reflected in the success rate of hospitals as mentioned later in the paper. When asked how long the respondent organization took to fully implement TQM, 72% indicated it is a continuing process, indicating that 28% believed that their TQM program had a limited life.

The questionnaire measured three indicators of a successful TQM program:

- Customer research
- Improved product
- Respondent assessment of success.

Total responses to these indicators reveal 34% have positive customer research, 38% partially positive, 1% not positive, and 27% not doing customer research. Improvement in product saw more positive results with 57% indicating improved product quality, 39% partial improvement, and 4% saying no improvement.

The last measure of success revealed that 45% of the respondents indicated that their TQM program is successful, while 50% say it is partially successful and 5% say it is not successful.

Tables 4, 5, and 6 show a statistical summary of individual TQM performance by type of organization.

A significant finding is the high rate of organizations that are not engaged in customer research. Manufacturing firms have the highest proportion for lack of customer research, hospitals the second highest, and service organizations the lowest. This finding is significant because of the fact that TQM's basic philosophy is determining the needs of the customer and meeting those expectations.

All three sectors show similar occurrences for being an unsuccessful experience. The tables also reveal a low occurrence of abandonment of TQM by all organizations.

In all three indicators of success, manufacturing firms show the greatest level of success while service organizations are the second most successful, and hospitals are the least successful. Furthermore, the tables reveal that hospitals have the highest occurrence of being partially successful, service organizations the second highest, and manufacturing firms the least partially successful.

The next series of tables compares qualitative and quantitative TQM tools by four comparisons. Table 7 summarizes the entire sample by the observed mean and percent time the tool is used. The least often used tools are affinity diagrams and selection grids. These two tools are used for the purpose of problem identification and solution planning. The reason that these two tools are not used as much may be the fact that several other tools are more popular and are designed for the same identical purpose. Some examples of the more popular tools are control charts, flow charts, brainstorming, and Pareto analysis. All of the tools fit into four categories, which are:

- Problem Identification

- Solution Planning
- Evaluating Indicators.
- Data Analysis

The statistical analysis later in the paper will discuss the analysis and results that were used to determine the extent to which either quantitative or qualitative tools were used.

In comparing manufacturing firms to the total sample, it is apparent that the use of qualitative tools is below average, and the use of quantitative tools is above average. Manufacturing firms have the highest level of success. The comparison of service organizations to the total sample shows a below average use of both qualitative and quantitative tools.

On the average, hospitals are using both qualitative and quantitative tools slightly more than the total sample. However, hospitals are the least successful. The two most used quantitative tools in the total sample are Pareto charts and sampling. As for qualitative tools, flow charts and brainstorming are the most significant. This average is consistent for each of the classification of organizations.

The questionnaire addressed the framework of TQM that has been developed from the Deming model. The PDCA Cycle, originally developed by Dr. Walter A. Shewhart, provides the structure for process improvement and Process Action Teams, which provides employee empowerment. Table 11 shows the extent to which these two factors are developed and used among all of the organizations.

Table 11 clearly indicates that organizations which are successful with TQM use Process Action Teams more than organizations which are not successful. Table 12 makes similar comparisons with the PDCA Cycle.

The low percentages shown in Table 12 are significant because of the fact that PDCA is such an important element of the framework of TQM. Hospitals use both Process Action Teams and the PDCA Cycle more often than manufacturing and service organizations since the accrediting organizations in 1994 are requiring their use. Organizations within the total sample that are successful use PDCA more often than organizations that are unsuccessful. Manufacturing companies and hospitals, which are successful, use PDCA more often.

Months of initial training was on the questionnaire to obtain how much training is given before the implementation of TQM. As can be seen in Table 13, most organizations train their employees an average of 5 to 6 months. Sixty-two percent of the service organizations spent over 6 months training their employees. A fairly significant percentage provided no training, which is obvious in the manufacturing sector.

Table 14, 15, and 16 show how often TQM tools are used by successful, partially successful, and unsuccessful organizations.

It is apparent, from a review of these tables, that organizations who are successful use quantitative tools more than organizations who are not as successful. The variance becomes greater for organizations that are not as successful. The only exception seems to be the qualitative tool of brainstorming, which is the most often tool used of the three groups (manufacturing, service, and hospitals). A major quantitative tool is control charts, which are used to identify special and common causes. Successful organizations use this tool 68% of the time.

Statistical Analysis

The statistical analysis consisted of a computation of the Spearman-rank correlation, hypothesis test, and logistic regression. The Spearman-rank correlation was used to compare the qualitative and quantitative TQM tools. This was done to determine whether a significant difference exists between the two populations of tools. The results of this computation revealed a correlation coefficient of -.65714. This coefficient suggests that there is a negative correlation between the two groups of tools showing that as the use of one group increases, the use of the other one decreases. This correlation was also used to calculate a Z statistic used for hypothesis testing.

The following hypothesis was formulated and tested:

H_0: There is no significant relation in the population between the Qualitative and the Quantitative tools. H_o: $\rho_s = 0$

H_A: There is a significant relation in the population between the Qualitative and the Quantitative tools. H_A: $\rho_s \neq 0$

The confidence level used to test the hypothesis was .05. With a two tail test, the acceptance region for Z is between ±1.96. The calculated z value equals -2.28; therefore, the null hypothesis is rejected causing the acceptance of the alternative hypothesis.

Logistic regression was conducted to determine if the qualitative and/or quantitative tools explain the variation of the dependent variable. Several choices of a dependent variable were used, including the question concerning whether the respondent considered their TQM program to be successful, the question of receiving positive customer research, and whether the product had improved. The logistic regression model expresses the dependent variable in binary terms of either "1" or "0". In this analysis, the "1" is unsuccessful, and the "0" is successful in each of the three categories in the table. Each of the dependent variables included both successful and unsuccessful responses. The purpose of using the logistic regression was to learn whether the qualitative or quantitative variables explained a variation in the dependent variables. Table 17 is a summary of the independent variables p-values obtain from the regression.

The overall p-values for each of the regression runs are as follows:

Stated Successful (Qualitative Tools)- .0004
(Quantitative Tools)- .0044
Improved Product (Qualitative Tools)- .0079
(Quantitative Tools)- .0009
Customer Research (Qualitative Tools)- .0002
(Quantitative Tools)- .0556

Below is a listing of the percentage of the variables correctly classified:

Stated Successful (Qualitative Tools)- 89.51%
(Quantitative Tools)- 90.60%
Improved Product (Qualitative Tools)- 92.13%
(Quantitative Tools)- 92.82%
Customer Research (Qualitative Tools)- 99.05%
(Quantitative Tools)- 96.12%

Both of the overall p-values and the percentages correctly classified indicate that regression model is significantly explaining the variation in the dependent variables. The only potential problem in p-values is with customer research (Quantitative Tools).

Conclusions

The sample of the population that was the focus of the study (Louisiana, Texas, Arkansas, Mississippi) suggests that the TQM management philosophy is very popular in this region of the United States. This finding is consistent with current literature, reporting that TQM is not only popular in the United States but also internationally.

The success rate of 51% for manufacturing, 40% service, and 34% for hospitals are surprising. The researchers expected the success rate to be much higher. The statistics gathered from the research show that many organizations are partially successful. This fact was not expected by the researchers. Also not expected was the low proportion of organizations using the PDCA Cycle since it is well recognized to be an important framework of TQM. Additional research to discover the reasons, structure, and their effectiveness is an interesting source of further research. Another interesting topic to explore in the future is the correlation of time since TQM is implemented to the success of the program. This could have positive implications on a large number of partially successful responses received.

Analysis of the data reveals that quantitative and qualitative tools do effect the success of a TQM program. It is quite apparent from Tables 14, 15, and 16 that quantitative tools are used more by organizations that are more successful. Also, as an organization moves from partially successful to unsuccessful in the study, quantitative tools became less used. Qualitative tools are also useful within the TQM philosophy, especially within brainstorming. Brainstorming is the number one used tool by all organizations in the study.

It was revealed in the Spearman-rank correlation that quantitative and qualitative tools are negatively correlated. This suggests that organizations concentrate on one group of tools at the expense of the other group. Tables 14, 15, and 16 support this finding. The hypothesis test also proved there was a significant difference between the two populations. The final test of Logistic regression further proved the relationship of the two groups of tools and the success of TQM.

Bibliography

1. Bhalla, Sushil K. and Satish K. Wason. "Managing the Technological Process." Quality Progress. January 1994: 81-85.

2. Braunstein, David. "Recapturing the Spirit." Journal for Quality and Participation. September 1989: 48-53.

3. Caudron, Shari. "Change Keeps TQM Programs Thriving."Personnel Journal. October 1993: 104-107.

4. Danne, Denise Jane. Total Quality Management in Industry and Its Implications for Secondary Education (Education Restructuring). Pepperdine University. 1992: 249.

5. Davies, Ken and Peter Hinton. "Managing Quality in Local Government and the Health Service." Public Money and Management. January-March 1993: 51-54.

6. Doyle, Kevin. "Who's Killing Total Quality?" Incentive. August 1992: 12-19.

7. Esler, Bill. "Teamwork Key to SPC Success." Graphic Arts Monthly. May 1992: 78-82.

8. Gilbert, James D. "TQM Flops - A Chance to Learn from the Mistakes of Others." National Productivity Review. Autumn 1992: 491-499.

9. Ginnodo, Bill and Richard S. Wellins. "Research Shows that TQM is Alive and Well." Tapping the Network Journal. Winter 1992-1993: 2-5.

10. Grayson, Curt. An Exploratory Study of the Relative Impact of Organizational Contextual Factors on Individual Decisions to Buy-In and Maintain Commitment to an Organization-Wide Total Quality Management Effort (Employee Satisfaction,

Recognition). California School of Professional Psychology. 1993: 247.

11. Heinrich, George. "Integrating TQM With Statistical and Other Quantitative Techniques." National Productivity Review. Spring 1994: 287-295.

12. Kim, Pan S. and Delinda D. Johnson. "Implementing Total Quality Management in the Health Care Industry." Health Care Supervisor. March 1994: 51-57.

13. Lengnick-Hall, Mark L., George Heinrich, and Earl Middleton. "Employee Involvement Makes TQM Work." Personnel Journal. October 1993: 108.

14. Longenecker, Clinton O. and Joseph A. Scazzero. "Total Quality Management from Theory to Practice: A Case Study." International Journal of Quality and Reliability Management. 1993: 24-31.

15. Pascoe, Larry Bruce. A Study of the Importance of Key Components of Total Quality Management Programs in American Manufacturing Firms (Quality Management, Management Commitment, Customer Orientation, Employee Involvement). United States International University. 1992: 156.

16. "The Quality March (Part 3)." Hospitals and Health Networks. January 5, 1994: 45-48.

17. Rohan, Thomas M. "New Crisis in Quality." Industry Week. October 15, 1990: 11-14.

18. Sorensen, Peter and Deborah Weber. "Organizational Cultural and TQM Implementation." Training and Development. April 1994: 69-71.

Tables

Table 1

TQM Research Total Sample

Type of Organization	Frequency	Relative Frequency
Manufacturing	184	.556
Service	55	.166
Hospital	92	.278
Total Useable	**331**	**1.00**

Table 2

Implementation of TQM

Type of Organization	Fully	Percent	Partially	Percent
Manufacturing	116	63%	68	37%
Service	36	65%	19	35%
Hospital	55	60%	35	38%
Total	202	56%	121	33%

Table 3

TQM Research Total Sample

Measures of Success	Yes	Partially	No	Not Doing
Customer Research Positive	34%	38%	1%	27%
Product Has Improved	57%	39%	4%	N/A
Stated TQM Successful	45%	50%	5%	N/A

Table 4

Customer Research Positive

Type of Organization	Yes	Partially	No	Not Doing
Manufacturing	41%	29%	1%	28%
Service	37%	37%	2%	22%
Hospital	16%	50%	2%	25%

Table 5

Product Has Improved

Type of Organization	Yes	Partially	No
Manufacturing	63%	31%	5%
Service	51%	44%	4%
Hospital	42%	48%	3%

Table 6

Stated TQM Successful

Type of Organization	Yes	Partially	No	Abandoned TQM
Manufacturing	51%	43%	5%	5%
Service	40%	56%	4%	7%
Hospital	34%	54%	5%	4%

Table 7

TQM Research Total Sample

Qualitative Tools	Mean	% Time Used	Quantitative Tools	Mean	% Time Used
Flow Chart	3.4	60%	Control Chart	3.3	58%
Cause & Effect Diagram	2.7	43%	Scatter Diagrams	2.4	35%
Multi-voting	2.4	35%	Pareto Charts	3.5	63%
Affinity Diagram	1.6	15%	Sampling	3.7	68%
Brainstorming	4.0	75%	Run Chart	3.1	53%
Selection Grids	1.5	13%	Histogram	3.3	58%
Task List	3.3	58%			

Table 11

Process Action Team				
Type of Organization	Mean	Percent of Time Used	Successful With TQM % Time Used	Unsuccessful With TQM % Time Used
Total Sample	3.8	70%	78%	63%
Manufacturing	3.7	68%	78%	55%
Service	3.7	68%	73%	63%
Hospital	4.2	80%	85%	82%

Table 12

PDCA Cycle				
Type of Organization	Mean	Percent of Time Used	Successful With TQM % Time Used	Unsuccessful With TQM % Time Used
Total Sample	2.5	38%	35%	30%
Manufacturing	2.3	33%	35%	22%
Service	2.1	28%	17%	25%
Hospital	3.0	50%	55%	50%

Table 13

Months of Initial Training						
Type of Organization	1-2	3-4	5-6	Over 6	No. Training	Mean
Manufacturing	19%	15%	7%	43%	11%	5-6
Service	11%	16%	4%	60%	5%	5-6
Hospital	15%	14%	14%	39%	4%	5-6

Table 14

Stated TQM Successful			
TQM Tool	Mean	Proportion Used	Quantitative Tool
Brainstorming	4.2	80%	no
Sampling	3.9	73%	yes
Pareto Chart	3.8	70%	yes
Control Chart	3.7	68%	yes
Histogram	3.6	65%	yes
Flow Chart	3.6	65%	no
Run Chart	3.5	63%	yes
Task List	3.4	60%	no
Cause & Effect Diag	3.0	50%	no
Scatter Diag.	2.7	43%	yes
Multi-Voting	2.4	35%	no
Affinity Diag.	1.7	18%	no
Selection Grid	1.7	18%	no

Table 15

Partially Successful			
TQM Tool	Mean	Proportion Used	Quantitative Tool
Brainstorming	3.9	73%	no
Sampling	3.6	65%	yes
Flow Chart	3.3	58%	no
Pareto Chart	3.3	58%	yes
Task List	3.2	55%	no
Histogram	3.1	53%	yes
Control Chart	3.0	50%	yes
Run Chart	2.9	48%	yes
Cause & Effect Diag	2.6	40%	no
Multi-voting	2.4	35%	no
Scatter Diag	2.2	30%	yes
Affinity Diag	1.6	15%	no
Selection Grid	1.5	13%	no

Table 16

Stated Unsuccessful			
TQM Tool	Mean	Proportion Used	Quantitative Tool
Brainstorming	3.7	68%	no
Sampling	3.0	50%	yes
Pareto Chart	2.8	45%	yes
Task List	2.7	43%	no
Control Chart	2.7	43%	yes
Histogram	2.6	40%	yes
Flow Chart	2.5	38%	no
Cause & Effect Diag	2.3	33%	no
Scatter Diag	1.9	23%	yes
Multi-voting	1.9	23%	no
Run Chart	1.8	20%	yes
Affinity Diag	1.2	5%	no
Selection Grid	1.0	0%	no

Table 17

Logistic Regression - Independent Variables p-values			
TQM Tools	Stated Successful	Improved Product	Positive Customer Research
Quantitative			
Control Chart	.993	.228	.800
Scatter Diag.	.626	.765	.702
Pareto Chart	.879	.961	.821
Sampling	.511	.089	.145
Run Chart	.020	.072	.202
Histogram	.516	.969	.371
Qualitative			
Flow Chart	.004	.010	.200
Cause & Effect	.990	.546	.122
Multi-voting	.494	.779	.620
Affinitive Diag	.692	.752	.178
Brainstorming	.843	.840	.971
Selection Grid	.985	.985	.985
Task List	.360	.554	.044

Senior Project Case Study: The Mentors' View

Kris Enzor
Kathi Gradle
K&C Small Business Computer Consultants, Inc.
kandc@usouthal.campus.mci.net

Abstract.

Mentors are recruited to assist senior project course teams who work on "real world" projects by helping to overcome obstacles in domain or technological areas. This paper attempts to describe, for one such project, the strategy, problems encountered, and the plans for improvement from the perspective of the mentors.

Introduction.

The School of Computer and Information Sciences at the University of South Alabama requires a two-quarter capstone project course sequence in the senior year. The sequence is intended to bridge the gap between academia and the "real world" by giving student teams an opportunity to work on projects that:

- involve greater complexity than those encountered in previous academic experiences
- involve "real world" situations and communication with "real" clients
- involve long-term commitment with the same team members for the same project.

The projects tend to be much larger in scope than previous projects and are often suggested by local industry or other university departments. Students are expected to devote sufficient time to successfully complete these projects and to treat their project as a "real" job. Students are expected to produce feasibility studies, meeting reports and journals, well-documented phases of the software development lifecycle and formal presentations.

During the past year, a project to redesign an existing software system involved ten students on three teams and two project mentors. The mentors, University alumni and consultants in a local computer consulting firm, presented the project to the Senior project class for the following reasons:

- Many of their projects are small to medium sized projects and therefore would be excellent learning experiences.
- The projects could lead to employment opportunities for students.
- It would be an opportunity for the alumni to share professional experiences with students.
- It would be an opportunity for the mentors to keep abreast of new philosophies and methodologies being taught in the University.

The mentors had the following goals and objectives planned, in accordance with the course objectives:

- To provide students to "real world" system design and implementation experiences.
- To provide opportunities to develop communication and interaction skills with "real world" clients.
- To complete a system implementation for a "real world" client.
- To deliver a well-documented, well-written system to the client.

The Strategy.

The client, a manufacturing facility, presently use a ten year old, custom software system to produce estimates and contracts for construction. The software lacks the functionality to maintain inventory, calculate labor costs, or produce shipping tickets. Many of these tasks are done through other software packages. The new system would extend the old system to include the order processing, inventory, and shipping functions while retaining all existing functionality.

The present system has no existing documentation other than source code and output reports. It was written using an obsolete database management system. Minor report changes had been accomplished over the past several years, but the addition of new functions was impossible to complete in a timely and cost effective manner.

The traditional "waterfall" development lifecycle was used in the analysis and design of the new software system. The mentors prepared a general plan for conversion to a new system consisting of the following five phases:

- Phase One: reverse engineer the present system, documenting its functionality, data requirements, and outputs.

- Phase Two: complete the analysis and design of the modules of the new system to meet existing functionality.
- Phase Three: extend the analysis and design to include new functionality and outputs according to the clients' request.
- Phase Four: develop, test, and install the new system modules to perform current functions.
- Phase Five: develop, test, and install additional modules, and complete user documentation.

The estimated time to complete these five phases was approximately twenty seven weeks. Phase One was scheduled for the first nine weeks. Phases Two, Three and Four were scheduled for the second nine week period. Phase Five would be performed in the final nine week period.

Phase One.

Because the Senior project class is only two quarters long, it was decided to begin the first phase with students from a Software Engineering class. The mentors provided the students with a laptop computer with the existing software installed and samples of all output documents. Weekly meetings were scheduled to answer student questions. Since the objective of this phase was to understand the functionality of the current system, the students did not have direct contact with the client. The team was prepared to make suggestions for improving the design and the user interface. The software engineering team prepared a system document of the existing software, as well as data flow diagrams, entity relationship diagram, and a set of system specifications.

Phase Two.

The second phase began with six students in the first quarter of the Senior Project course. No student from the software engineering team belonged to this group. However, one was available for consultation. The new team received the existing system documentation from Phase One together with the laptop computer containing the existing software.

The objectives given to the team for completion in this phase were:

- to analyze the supplied information
- to conduct interviews with the mentors and the client
- to produce a high level design specification for the order processing, shipping, inventory and scheduling modules, and
- to produce a low level design specification for the modules to replace the existing system

These six students worked together for several weeks analyzing the business requirements, conducting interviews and preparing design documents. At this point, they separated into two teams, consisting of three students. One team, the design team, moved to Phase Three and the other, the implementation team, to Phase Four.

Phase Three.

The requirements of this phase included low level definition, analysis and design of the additional modules. This team was provided with source documents supporting order processing, inventory, shipping and scheduling operations which were to be automated. The resulting analysis and design document was to be added to the existing documentation to provide the total system view.

Phase Four.

The implementation team was assigned the software development process. User interface screens were designed for approval by the client. A working version of the software, with user documentation, providing the current functionality was to be delivered.

Phase Five.

The design and implementation teams were to rejoin to extend the system development in Phase Four by the functions identified in Phase Three. Final versions of the end user reference manuals and system documentation were to be completed. The new system would be installed and training provided to the users. The students would have the opportunity to monitor the installation for satisfactory performance.

The Problems Encountered.

The major problems experienced during the project included:

- misdirected focus
- ineffective communication
- poor group interaction skills
- inadequate time management skills
- lack of commitment
- unrealistic expectations

The problems stated above are interrelated and indicative of the problems seen in the "real world". They affected the mentors, the students, and, ultimately, the success of the project.

The focus of the project shifted, prematurely, from analysis and design issues of Phase Two to the implementation issues of Phase Four. The team did not fully understand the document they received from Phase One. They relied on it as the complete analysis and design of the new system. This document was not the analysis and design for the new functions of the system, only those of the existing system.

Their concerns shifted to the implementation tools they were required to use for programming. Team anxiety over using an unfamiliar tool translated to a focus on the tool, not the problem. Time was lost trying to redirect the focus.

At this point, communication problems began to arise. First, the mentors were failing to communicate with the students. The students did not regard the mentors as resources to help in the organization of the tasks to be performed. Instead, the students reacted to requests for documentation revisions and time logs as obstacles to their progress and unreasonable demands. The students were also experiencing communication problems within their teams. Some team members regarded each other with mistrust. The course professor attempted to reestablish dialog between all parties involved, but had little success.

Poor communication naturally led to poor group interaction. Once the team split into two groups for Phases Three and Four, the two teams halted all interaction with each other and developed their own agendas. The team members failed to realize that one team could not succeed without the other. The final deliverables depended on the combined efforts of the two teams.

Time management became a problem early in Phase Two. The students did not understand the type of commitment necessary to complete a project of this scale. Initially, the team was allowed to set their own schedules, but as it became clear the students were inexperienced at project planning, the mentors sought to help them set deadlines.

Another time management issue developed when the project team compared the status of their project with other projects in the class. Their classmates were in the requirements development phase. This project was at a different stage in the lifecycle, because it had begun a quarter earlier. Therefore, it could not be compared to the other projects. The students expended much time and energy on activities unrelated to their project.

Time management was further compromised by the fact that some students were enrolled in the Systems Analysis and Design course simultaneously with the Senior Project course. These were students caught in catalog transitions because of curriculum changes. Both courses are very time consuming. In addition, analysis and design skills are essential to the Senior project course.

Long-term commitment to the project was missing because a majority of the team members were not required to enroll in both quarters of the Senior Project course due to curriculum changes. The mentors perceived a lack of commitment from some of the team members due to their personal obligations. They were not committed to their project as a "real job".

To the mentors, this was a "real job". The client agreed to participate at the request of the mentors, at no cost to the client. All work done by the students was considered part of their regular course work, the mentors donated their time to the project. A client bill was produced detailing the hours worked, but no charge was associated with the hours. The mentors made a commitment to the client to deliver a working, documented system.

It became apparent that the expectations were unrealistic under the current conditions. The original plan and time frame could easily have been met with teams comprised of experienced systems analysts. With the additional time required to assist inexperienced analysts in the learning process, the project was still feasible. The unfortunate communication and interaction problems undermined the analysis and design phases of the project.

Plans for Improvement.

Some areas for improvement include:

- documentation
- client communications
- mentor communication
- group interaction skills

Better documentation requirements should be prepared for the students. The students were unable to produce adequate documentation because, they felt, they had learned several different standards in different classes. They did not know how to achieve the documentation that was specified. Hopefully, by more clearly stating the specification, the documentation will be easier to produce. Because the same team might not follow the project to completion, communication through documentation is critical. This could be accomplished by reviewing previous documentation and stressing the value to the user and the programmer.

In order to assist the students in client interviews, mock interviews would be required to insure the students are prepared for client interaction. The students could practice expressing the problem statement using client terminology while developing appropriate questions for the client. The mentors would act as the client in these mock interviews in order to establish better communications with the students.

The mentors would require more student contact through written and verbal status reports. Each team member would have an area of responsibility and report on their progress individually and as a team. These reports would be brief and in journal format including date, amount of time, and activity performed. By reviewing these reports frequently, the mentors would be able to maintain the focus of the project and identify potential time management and communication problems.

Documentation, mock interviews and more student contact will promote the opportunity for better group interaction. Planning more working meetings would provide time for the students to get help for specific problems they may have. The mentors would like to share other aspects of systems development such as hardware installation, exploring development tools, network installations, or reviewing other projects.

Conclusion.

After a careful review of this project, the mentors have determined there were flaws in their strategy, which led to some of the problems encountered. Many of the problems can arise in any project. Time management and communication are critical to all projects. The mentors will continue to improve their strategy and skills and look forward to working with another Senior project team in the future.

Exploding the Myth that the Gender Factor is a Significant Determinant of Performance in Computing Courses at a Four-Year College

Alexander Heslin, Jr., Ph.D
Department of Computer Technology
Fort Valley State College

Abstract

The purpose of the study is to determine whether there are significant differences in performance among males and females in the Computer Literacy course given at a four-year historically black college. Also, the research was conducted to determine the influence, if any, of male versus female instructors. The class is required for many of the majors in the School of Arts and Sciences, School of Agriculture, and the School of Education. A typical Computer Literacy class at this college is composed of at least .98 African-American students. The class consists of intensive lecture and lab work in five modules: Computer concepts (hardware and software architecture, programming and systems analysis concepts, robotics, artificial intelligence, and societal impact), operating systems, word processing, spreadsheets, and database technology. Classes meet each day and students are required to turn in approximately 50 lab practicals during the quarter. The results of the study are based on the performance of 875 students over a four (4) year period.

Summary

The purpose of the study is to determine whether there are significant differences in performance among males and females in the Computer Literacy course given at a four-year historically black college. Also, the research was conducted to determine the influence, if any, of male versus female instructors. The class is required for many of the majors in the School of Arts and Sciences, School of Agriculture, and the School of Education. A typical Computer Literacy class at this college is composed of at least .98 African-American students. The class consists of intensive lecture and lab work in five modules: Computer concepts (hardware and software architecture, programming and systems analysis concepts, robotics, artificial intelligence, and societal impact), operating systems, word processing, spreadsheets, and database technology. Classes meet each day and students are required to turn in approximately 50 lab practicals during the quarter. The results of the study are based on the performance of 875 students over a four (4) year period.

Background

Is there really a significant difference in performance among male and female students in computer courses? Are female students more anxious in working with computers? Do female students have more negative attitudes toward computers? Do male students have more foresight about the use of computers in their future occupations? The popular belief is that the answer to these and other questions regarding female use of computers is "yes." Although the commonly-held beliefs degrade the image of females, observations and performance records in the classroom fail to support the myths.

The myths originate in published studies comparing performance levels of males and females in elementary and secondary schools and are cited in gender research conducted at the college-university and workplace levels. Often, the studies report marginal levels of statistical significance that is overlooked by the next generation researcher citing the work. The result is that the original study is taken out of context and accepted as though there is true significant difference (Denmark, Russo, Frieze,and Sechzer, 1988). The next generation researcher may perpetuate the myth by publishing more marginally statistical significant work based on marginally statistical significant work, and so forth. Eventually, the compound error is publicized in the media and becomes crystallized as the salient beliefs of parents, students, and personnel departments. Valerie Clarke (Clarke,1992,p.73) summarizes the process:

> In those cases where parents have read these exaggerated reports (e.g., Benbow and Stanley, 1980) they have accepted them and believed that their daughters' lack specific abilities, whereas parents who have not read these reports believed their sons' and

daughters' abilities were equivalent (Eccles and Jacobs, 1986). Essentially, the small differences reported in the research have lead to the development of sex-based differences in the attitudes and expectations, subject choices, and performance outcomes of their sons and daughters.

Prior Research

The literature is rich in studies relating gender and computer use. For example, the ERIC database on CD-ROM contains over 200 entries.

Research Studies Supporting Gender Differences

Most of the studies reviewed for this work support the gender myths. For example, an international study (Reiner and Plomp, 1994) involving over 70,000 students in 10 countries concluded that females know less about information technology than males, enjoy using the computer less, and expect more software problems. Students with positive attitudes toward computers have greater expectations that their future occupation will require the use of
computers and females have less expectations than males (Nickell, 1987). Wilder et al (Wilder, Mackie, and Cooper, 1985) report that students describe computers as "masculine" objects, suggesting that students approach computer interactions as though the computer is "male." In the same study entering college students were surveyed, and it was found that females who had computing experiences comparable to males nonetheless felt less experienced than males. Webb (Webb, 1984) reports that female-majority groups showed lower achievement than males and that the females directed most of their interaction to the males. In that study, Webb found that females are less likely to obtain any needed help in a mixed-sex group. Females tend to defer to males in cases where both want to use the computer (Boss,1982). Females, as a group, rate themselves lower than males for perceived self-efficacy on computers (Dambrot, Watkins-Malek, Silling, Marshall and Garver, 1985; Lockheed, 1985; Miura, 1987; Ogozalek, 1989). Females see computers as tools to solve problems (Gerver, 1989;Marko of, 1989). When they find that they can't solve problems with computers, females lose interest in them (Bernstein, 1992). In a study of corporate work environments (Michaels, 1993), results of statistical analysis indicated significant gender differences on measures of computer anxiety and overall attitude toward computers.

Research Studies not Supporting Gender Differences

Other studies have found contradictory results. Females have a less positive attitude toward computers than males, but the difference in attitude is not significant (Nickell, 1987). Females are far more pleased than males with success at the computer and this in turn fosters a positive attitude (Aman, 1992). Females enrolled in accelerated math classes rejected the notion that computers are a male domain, felt that computers would be helpful to them in the future, and were positive about their computer studies (Kwan, 1985). Other studies demonstrate that females can compete (or outperform) their male counterparts (Anderson, 1987; Clarke and Chambers, 1989; Fetler, 1985; Ivenstosch, Kersteen, and Linn, 1987; Linn, 1985). Women who work with computers outnumber men more than two to one (Losee, 1992). Childers (Childers, 1992) found no indication that feminine sex-typed females were more negative about computers (although masculine sex-typed males had significantly more negative computer attitudes than androgynous males or females). Childers (1992), Grogan (1991), Sariya (1991), Carl (1988), Von Holzen (1993) concluded in their doctoral dissertations that gender did not predict computer anxiety and attitudes toward computers. Baker (1990) found that the relationship between the level of computer anxiety and the gender, the age, or the racial heritage of her study group was not significant.

Research that Questions the Validity of Gender Differences Studies

Kay(1992) found a great deal of confusion in research of gender differences in computer-related behaviors. She recommends that researchers be more specific about what attitudes are being measured, be more precise about the skills being measured, specify the tasks the computer is being used for, and what age group is in the sample.

Summary of Prior Research

As seen above, a large percentage of the research to date has concentrated on linking attitudes and gender. here are studies (e.g. Anderson, 1987; Clarke and Chambers, 1989) that correlate gender and behavior (i.e., performance). Unfortunately, the popular image of females and computing has come from the studies linking gender and attitudes. The problem is that attitudes are not a stable characteristic and do change over time. Many researchers have found that attitude toward computers (or any information technology) changes with increased usage, regardless of gender. Therefore, measuring attitudes is similiar to hitting a moving target. What is needed is more research confirming or denying the popular belief that females **perform** less well than males at computing.

The Research Study

In the past four years, 875 students have enrolled in the Computer Literacy course. The performance records of the students were analyzed using LOTUS 1-2-3 and MINITAB. The probability distribution of the grades for this period of time is as shown in Table 1.

The statistics on the percentage of students withdrawing from the Computer Literacy class are misleading. A number of students pre-enroll in the course in order to have the minimum number of credit hours required by Financial Aid. These students usually drop the class without ever attending. Table 1 demonstrates that female students perform better than their male colleagues at every grade level.

	A	B	C	D	F	I	W
Males	.103	.257	.215	.151	.125	.029	.119
Females	.179	.264	.245	.101	.092	.025	.093

Table 1. Distribution of Final Grades (Including Incompletes and withdrawals)

Tests for Significant Differences

Statistical tests were conducted to determine if there is a significant difference in performance among the male and female students data. The z test was used to test the equality of the proportions listed in Table 1 above. At alpha = .05 the hypothesis that there is no significant difference in the proportion of students who make A, B, C, D, F, I, and W was tested. The z values obtained from the tests are:

$Z_a = 2.10$	Significant
$Z_b = 0.19$	Not significant
$Z_c = 0.83$	Not significant
$Z_d = 1.16$	Not significant
$Z_f = 0.77$	Not significant
$Z_i = 0.09$	Not significant
$Z_w = 0.62$	Not significant

In summary, there is no evidence of a significant difference in performance among males and females at all grade levels, except in the "A" category. In the "A" category, females performed significantly better than male students.

Research indicates that female students perform better with female instructors. The performance data for the students, by sex of instructor, is shown in Table 2 (male instructors) and Table 3 (female instructors).Statistical tests were conducted to determine if there is a significant difference in the performance data of male and female students for male instructors and for female instructors.

The z test was used to test the equality of the proportions in Tables 2 and 3 above. At alpha = .05 the hypothesis that there is no significant difference in the proportion of students who make grades of A, B, C, D, F, I, and W was tested. The hypotheses are that male and female students perform equally for male instructors, and male and female students perform equally for female instructors.

	A	B	C	D	F	I	W
Males	.100	.246	.242	.137	.114	.041	.119
Females	.186	.276	.240	.097	.087	.031	.082

Table 2. Distribution of Grades by Sex (Male Instructors)

	A	B	C	D	F	I	W
Males	.109	.283	.152	.184	.152	.000	.120
Females	.159	.232	.258	.113	.106	.007	.126

Table 3. Distribution of Grades by Sex (Female Instructors)

The z values obtained for the **male instructors** are:

$Z_a = 2.00$ Significant
$Z_b = 0.71$ Not significant
$Z_c = 0.05$ Not significant

$Z_d = 0.80$ Not significant
$Z_f = 0.54$ Not significant
$Z_i = 0.19$ Not significant
$Z_w = 0.73$ Not significant

The z values obtained for the **female instructors** are:

$Z_a = 0.67$ Not significant
$Z_b = 0.68$ Not significant
$Z_c = 1.54$ Almost significant
$Z_d = 0.82$ Not significant
$Z_f = 0.47$ Not significant
$Z_i = 0.11$ Not significant
$Z_w = 0.08$ Not significant

Contradictory to expected, male instructors give their female students significantly more grades of "A", while female instructors do not. The results indicate that there is no significant difference in the grades of male and female students as given by female instructors.

Conclusions

The results of this study show that for a large population of students there are basically no significant differences in the performance of male and female students in the computer literacy course, with the exception that females receive significantly more grades of "A" than males. The results also show that the gender of the instructor makes little difference in the distribution of final grades, with the exceptions that female students achieve the final grade of "A" significantly more than males when taking the course from male instructors and female instructors give the final grade of "C" at an almost-significant level higher to females than to males.

References

Aman, J.R. (1992) Gender and attitude toward computers. In Search of Gender Free Paradigms for Computer Science Education, Edited by C. Dianne Martin and Eric Murchie-Beyma, International Society for Technology in Education Publications, 33-46.

Anderson, R.E. (1987). Females surpass males in computer problem solving: Findings from the Minnesota computer literacy assessment. Journal of Educational Computing Research, 3(1), 39-51.

Baker, N.T. (1990). The development and validation of an instrument to measure computer anxiety in adult learners in technical colleges. Ph.D. dissertation. University of Wisconsin - Madison. pp:207.

Benbow, C.P. and Stanley, J.C. (1980). Sex differences in mathematical ability:Fact or artifact? Science, 210, 1262-1264.

Bernstein, D.R. (1992). A new introduction to computer science. In Search of Gender Free Paradigms for Computer Science Education. Edited by C. Dianne Martin and Eric Murchie-Beyma, International Society for Technology in Education Publications, 87-91.

Boss, J.A. (1982). Sexism among the micros. The Computing Teacher, 11(8), 10-12.

Carl, D.L. (1988). The impact of educational level, gender, age, and computer experience on computer attitudes of adults enrolled in ABE and GED programs in the state of Arkansas. Ph.D. dissertation. University of Arkansas. pp:217.

Childers, L.W. (1991). Selected predictors of educational Administators' computer anxiety and attitudes toward computers. Ph.D. Dissertation, University of New Orleans pp:264.

Clarke, V. A. (1992). Strategies for involving girls in computer science. In Search of Gender Free Paradigms for Computer Science Education. Edited by C. Dianne Martin and Eric Murchie-Beyma, International Society for Technology in Education Publications, 71-86.

Clarke, V.A. and Chambers, S.M. (1989). Gender-based factors in computing enrollments and achievement: Evidence from a study of tertiary students. Journal of Educational Computing Research, 5, 409-29.

Danbrot,F., Watkins-Malek, M., Silling, S.,Marshall,R. and Garver,J.(1985). Correlates of sex differences in attitudes toward and involvement with computers. Journal of Vocational Behavior, 27, 71-86.

Denmark,F., Russo, N.F., Frieze, I.H. and Sechzer, J.A. (1988). Guidelines for avoiding sexism in psychological research: A report of the ad hoc committee on nonsexist research. American Psychologist ,43, 582-585.

Eccles,J.S. and Jacobs,J.E. (1986).Social forces shape math attitudes and performance. Signs,11,367-389.

Fetler, M. (1992). Sex differences on the California statewide assessment of computer literacy. Sex Roles , 13, 181-92.

Gerver,E. (1989). Computers and gender. In T.Forester(Ed.) Computers in the Human Context, pp.481-501. Cambridge,MA:MIT.

Grogan, V.C. (1991). Computer Attitudes of selected students and educators in relationship to computer access and experience and gender. Ph.D. dissertation. University of Nebraska-Lincoln.

Hill,T.,Smith,N. and Mann,M. (1987). Role of efficacy expectations in predicting the decision to use advanced technologies: The case of computers. Journal of Applied Psychology, 72, 307-313.

Iventosch, L., Kersteen, Z.A. and Linn, M.C. (1987). An exploration of gender-related study skill differences in computer science. Paper presented at the Annual Meeting of the American Educational Research Association. Washington, D.C., April.

Kay, R.H. (1992). An examination of gender differences iin computer attitudes, aptitude, and use. Paper presented at the Annual Conference of the American Educational Research Association (San Francisco, CA), April 20-24, 28p.

Konvalina,J., Wileman,S. and Stephens,L. (1983). Math profiency: A key to success for computer science students. Communications of the ACM, 26(5), 377-382.

Kwan, S.K. (1985, September). Gender differences and computing: Students' assessment of societal influences. Education and Computing , I, 187-94.

Linn, M. (1985). Gender equity in computer learning environments. Computers and Social Sciences, I,19-27.

Lockheed,M. (1985). Women,girls and computers: A first look at the evidence. Sex Roles, 13(3/4),115-122.

Losee, S. (1992). Surprise! A PC gender gap. Fortune, 7/13/92, Vol. 126, Issue 1, p14.

Markoff,J. (1989,February 13). Computing in America: A masculine mystique. New York Times, pp.A1,B10.

Michaels, M.C. (1993). Individual differences on measures of computer aversion in corporate work environments (human computer interaction). Ph.D.dissertation. University of Minnesota. pp:123

Miura,J.P. (1987). The relationship of computer self-efficacy expectations to computer interest and course enrollment in college. Sex Roles, 16(5/6),303-311.

Nickell, G.S. (1987). Gender and sex role differences in computer attitudes and experience. Presented at the Annual Convention of the Southwestern Psychological Association (33rd, New Orleans LA, April 16-18)

Ogozaled,V. (1989). A comparison of male and female computer science students' attitudes toward computers. SIGCSE Bulletin (Special Interest Group on Computer Science Education of the ACM),21(2),8-14.

Reiner, I.J. and Plomp, T. (1994). Computer use: another area of inequity? Paper presented at the Annual Meeting of the American Educational Research Association (New Orleans, LA, April 4-8) that:

Sariya,Y. (1991). A study of college students' attitudes toward computers. Ph.D. Dissertation. University of Missouri - Columbia, pp: 137.

Von Holzen, T. (1993). Effects of group composition and gender on college students' computer knowledge and attitude. Ph.D. dissertation. Texas Tech University, pp:150.

Webb, N.M. (1984). Sex differences in interaction and achievement in cooperative small groups. Journal of Educational Psychology, 76(1), 33-44.

Wilder, G. Mackie, D.M. and Cooper, J. (1985). Gender and computers: Two surveys of computer-related attitudes. Sex Roles, 13, 215-228.

PANEL
ISECON '96

NEW ACCREDITATION STANDARDS OF THE AMERICAN ASSOCIATION OF COLLEGIATE SCHOOLS OF BUSINESS

MODERATOR AND PANELIST:
C. Somarajan, Associate Professor
Department of Management
Harrison College of Business
Southeast Missouri State University
Cape Girardeau, MO 63701
(573) 651-2900 or FAX (573) 651-2992
c833bum@semovm.semo.edu

PANELISTS:
Dean Gerald McDougall and Stanley Stough
Southeast Missouri State University

In 1991 AACSB (American Association of Collegiate Schools of Business) revised its standards for accreditation which were "research-driven." The new standards are mission-driven, which enable more schools not as highly research-oriented, to go for accreditation. Harrison School of Business at Southeast Missouri State University, Cape Girardau, Missouri, went for accreditation under the new standards.

We went through the pre-candidacy stage in 1994 and submitted our self-evaluation report for accreditation in Fall 1995. The AACSB accreditation team visited the school in February 1996. The AACSB Accreditation Board, after their April 23, 1996 meeting, officially announced the accreditation for the Harrison College of Business, Southeast Missouri State University.

The authors were closely associated with the accreditation process as the College Dean, Chair, and member of many committees, such as Faculty Composition and Development and Strategic Planning, We worked together to conduct the College's self-evaluation. The topic may be of interest to many colleagues at the conference. We will discuss the processes and the issues and how we got the accreditation.

ISECON '96

INDEX OF SPEAKERS