ISECON '97

INFORMATION SYSTEMS EDUCATION CONFERENCE

PROCEEDINGS

THE WONDERFUL WORLD OF INFORMATION SYSTEMS

OCTOBER 17-19, 1997
ORLANDO, FLORIDA

SPONSORED BY THE FOUNDATION FOR INFORMATION TECHNOLOGY EDUCATION
THE RESEARCH AND DEVELOPMENT ARM OF AITP

PROCEEDINGS

ISECON '97

INFORMATION SYSTEMS EDUCATION CONFERENCE

THE WONDERFUL WORLD OF INFORMATION SYSTEMS

Orlando, Florida
October 17-19, 1997

Marvin Albin, Editor
Dakota State University

© 1997
Foundation For Information Technology Education
THE ASSOCIATION OF INFORMATION TECHNOLOGY PROFESSIONALS

FOUNDATION FOR INFORMATION TECHNOLOGY EDUCATION
EDUCATION FOUNDATION
Associated with Association of Information Technology Professionals
505 Busse Highway, Park Ridge, IL 60068-3191
(847) 825-8124 Phone (847) 825-1693 Fax
http://www.edfoundation.org

T.M.

Board of Regents

President
Louis J. Berzai, CCP
Notre Dame, IN

Vice President
Alan Strong, CDP
Los Angeles, CA

Regent
Leo de Folter, CMA
Winnipeg, Canada

Regent
Alfred Jones
Palmyra, MO

Regent
Ralph Jones
Arlington, VA

Regent
Carol Lee Logan
Elizabethton, TN

Director of Finance
Jay Fitzgerald

Advisory Council
George Brenner
Vice President/CIO
MCA-Universal Studios

Paul Gillin
Editor-in-Chief
ComputerWorld

Max D. Hopper
President
Max. D. Hopper
Associates

Peter Karmanos, CEO
CompuWare
Corporation

J.T. Westermeier, CDP
Partner
Fenwick and West LLP

Dear Information Systems Educators

On behalf of the Foundation for Information Technology Education Board of Regents, it is with great pleasure that I welcome you to ISECON '97: "The Wonderful World of Information Systems", being held at Disney World, in Orlando, Florida. This is an excellent opportunity to learn from and share ideas with your peers regarding the field of Information Systems.

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.

We all hope that you enjoy your stay at the Disney's Coronado Springs Resort at the Walt Disney World Resort. This is a conference that includes the family. Hope everyone enjoys the facilities.

Be sure to attend the Friday night reception from 5pm until 6pm for the vendors that are participating. Also, please attend the reception for vendors and attendees of ISECON '97 on Saturday from 5pm until 6pm. We have tried to schedule the activities so that everyone can enjoy the facilities at Disney World. All of the elements of a fantastic educational experience is here for you.

Welcome to ISECON "97: The Wonderful World of Information Systems.

Best Wishes,

Lou

Louis J. Berzai,CSP
President
Foundation for Information Technology Education
Board of Regents

DAKOTA STATE UNIVERSITY
Madison, South Dakota 57042-1799

ISECON '97

The Wonderful World of Information Systems

On behalf of the Foundation for Information Technology Education, and the ISECON '97 committee, I welcome you to Orlando, ISECON 97,and the Wonderful World of Information Systems!

This year's ISECON conference promises to be as exciting as other ISECON conferences in the past. With the booming of the world wide web, we have many timely offerings about using the Internet for Information Systems Education. We have papers on all kinds of technology and curriculum issues. We have several exciting international papers and presentation. This ISECON will refresh you with zestful new ideas and methods to improve your information systems instruction!

Your participation in ISECON, the premier conference in IS education, is appreciated! For those that are presenting, thank you for taking the time to bring your knowledge to ISECON; for those that reviewed, thank you for assisting this conference; for those of you that are attending without prior participation, we know that you will consider submitting papers, workshop and panel ideas and serving in some capacity for ISECON '98 in San Antonio!

Take in the curriculum update on Friday morning and enjoy your copy of the updated Four-Year Information Systems Curriculum! Enjoy our exciting keynote speaker on Saturday! Meet old friends and make new friends at the receptions! Take time to meet the vendors! And take time to enjoy the Magical World of Orlando and Walt Disney World!

Sincerely,

Bruce

Bruce White
Conference Chair

ISECON '97
THE WONDERFUL WORLD OF INFORMATION SYSTEMS

LEADERSHIP TEAM

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

AITP-FOUNDATION FOR INFORMATION TECHNOLOGY EDUCATION
Dr. Lou Berzai
Notre Dame University
South Bend, Indiana

PROCEEDINGS
Dr. Marvin Albin
Dakota State University
Madison, South Dakota

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

ISECON '97
THE WONDERFUL WORLD OF INFORMATION SYSTEMS

THE PROGRAM COMMITTEE

PROGRAM CHAIR

Roy Daigle, University of South Alabama

TRACK CHAIRS

IS CURRICULA
William Tastle, Ithaca College
Rex Dumdum, Marywood College

WORKSHOPS
Marino Niccolai, University of South Alabama

PANELS
Michael V. Doran, University of South Alabama

SMALL COLLEGES, TWO YEAR OR COMMUNITY COLLEGE PROGRAMS
Pam Chapman, Waubonsee Community College

EMERGING AND LEADING EDGE TECHNOLOGIES
Ruth Guthrie, University of Redlands

INDUSTRY AND IS EDUCATION RELATIONSHIPS
Kevin Elder, Angelo State University

INTERNATIONAL IS ISSUES
Eli Cohen, Grand Valley State

ISECON '97
THE WONDERFUL WORLD OF INFORMATION SYSTEMS

REFEREES

John Barr, Ithaca College
France Belanger, U of San Francisco
Pam Chapman, Waubonsee Community College
Eli Cohen, Grand Valley State
Roy Daigle, University of South Alabama
Michael V. Doran, University of South Alabama
Rex Dumdum, Marywood College
Kevin Elder, Angelo State University
Irv Englander, Bentley College
George Fowler, Fowler@unix.tamu.edu, Texas A&M
Ruth Guthrie, University of Redlands
Susan Helms, Metro State College, Denver
Buzz Hensel - Univ. of Texas Arlington
Herman "Hoppy" Hoplin - Syracuse University
Kevin Hufford, Ithaca College
Ric Jentzsch, Uni of Canberra, Australia
Karlheinz Kautz, Norwegian Computing Center/Norsk Regnesentral, Oslo Norway
Janet Kemp, University of South Alabama
Gwynne Larsen, Metro State College, Denver
Joyce Currie Little, Towson State
H. E. Longenecker, University of South Alabama
Brian Lynch, Marywood University
Marcus Lynch, U of the West of England
Munir Mandalwala, Temple University
Marino Niccolai, University of South Alabama
John Olson, University of South Alabama
Patrick Olson, Claremont Graudate School
Joe Omolayole, Grambling State U.
William Owen, University of South Alabama
Robert Ryder, University of South Alabama
Donna Schaeffer, University of Redlands
William T. Schiano, Bentley College
Conrad Shayo, California State University San Bernadino
Jim Spee, University of Redlands
Susan Suchocki, University of LaVerne
William Tastle, Ithaca College
Sharon Vest, University of South Alabama
Les Waguespack, Bentley College
Patricia Woodworth, Ithaca College

ISECON '97
PROCEEDINGS

TABLE OF CONTENTS

Friday October 17, 1997

Saturday, October 18, 1997

IS ' 97 Completion, Publication and Deployment

Herbert E. Longenecker, Jr.: bart@cis.usouthal.edu
David L. Feinstein: feinstein@cis.usouthal.edu
School of Computer and Information Science
University of South Alabama, Mobile AL 36688

Overview

IS ' 97 represents the culmination of a four-year effort by the CoChairs, task forces and reviewers and stands completed as the first Model Curriculum for Undergraduate Programs of Information systems (Davis 1997). The work now represents the official model curriculum for the ACM, AIS and AITP (formerly DPMA). The initial work was presented in 1994 (Gorgone 1994) and in 1995 (Longenecker et al 1995) and has been presented at ISECON, IAIM, IACIS, SigCSE, DSI, and IRMA conferences.

During this panel discussion, the high points of the model will be presented. Emphasis will be placed on the learning units, and examples will be provided to show how the model curriculum can be utilized in any IS program, even without changing any course titles.

Finally, results will be presented and the initiative of the CoChairs to study the issue of developing an IS accreditation mechanism will be discussed. Commentary will be invited.

References

Davis, Gordon B., John T. Gorgone, J. Daniel Couger, David L. Feinstein, and Herbert E. Longenecker, Jr. 1997. IS=97: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. Published by AITP, Chicago 1997.

Gorgone, John T., J. Daniel Couger, Gordon B. Davis, David L. Feinstein, George Kasper, and Herbert E. Longenecker 1994. "Information Systems '95", DATA BASE, Volume 25, No.4 November 1994, pp 5-8.

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.

Using Use Case Modeling For Object-Oriented Analysis Education

Kevin C. Dittman, Assistant Professor
Dept. of Computer Technology, Purdue University
Lonnie D. Bentley, Professor
Dept. of Computer Technology, Purdue University

Many companies are embarking on the paradigm shift from using traditional structured analysis and design techniques for software development, to using object-oriented (OO) methods and techniques. As Information Systems educators, we must start teaching our students the object-oriented skills which companies demand, and provide an environment for them to effectively learn those skills. This workshop will present a method of using Use Case Modeling to teach object-oriented analysis in the classroom as well as in the laboratory. Use Case Modeling, developed by Dr. Ivar Jacobson, is a technique that has proven to be extremely successful in identifying and analyzing system requirements, as well as aiding in the identification of objects and their behaviors. So successful in fact, that it is being included as part of the Unified Modeling Language (UML), a new, future standard for OO modeling languages.

Workshop participants will be presented with a method for performing use case modeling, a demonstration of using a CASE tool to construct the various use case models, as well as the techniques and tools used in documenting the individual use cases. In addition, this workshop will present how use cases can assist in the identification of objects and their behaviors, plus how use cases can support other phases of the system development life cycle. All of the methods and tools presented in this workshop are currently being used in the Computer Technology systems courses at Purdue University.

Workshop Agenda:

A. Introduction
B. Object-Oriented Basics
   1. A Brief Introduction to Objects
   2. Object Oriented Methods
C. Overview of Use Case Modeling
   1. Use Case Modeling
   2. Using Use Case Modeling in the Classroom
   3. The Use Case Method
D. Building The Analysis Modeling using Use Case Modeling
   1. Determining Project Scope and Boundaries
   2. Identifying Primary Actors and Business Events (Use Cases)
   3. Building the Use Case Model
   4. Identifying Use Case Dependencies
   5. Documenting the Use Cases
E. Object Analysis
   1. Identifying Business Objects from Use Cases
   2. Identifying Object Behavior from Use Cases
   3. Performing Robustness Analysis
F. Where Do You Go From Here?
   1. How Use Cases Support Design
   2. How Use Cases Support Implementation
G. Closing

The Pace of Change in Technology and Its Ramifications on IS Curricula and Budgets

Dr. Marvin Albin
Dakota State University
e-mail: albinm@zeus.dsu.edu
3534 S. Western Avenue
Sioux Falls, SD 57105
605.367.5380
fax: 605.367/5383

Software version upgrades have been released in increasingly short cycles. What began with three- to four-year cycles moved to two-year cycles, then to one-year cycles. It is not unusual now for a major upgrade to occur in less than twelve months. These short cycles and the timing of the release can play havoc with curriculum planning regarding the incorporation of software in courses and in programs. Certainly no one expects that a student graduating from a program four years after entry would remain knowledgeable about current software from courses taken four years earlier, but what should be done to ensure that graduating students are at least somewhat familiar with current software upon graduation? What do employers of these graduates have a right to expect? Just how soon after release should new software versions be incorporated into a curriculum? Indeed, how soon can they given budget considerations? And, how are we ensuring that graduating students have knowledge or awareness current software?

Hardware advances occur as rapidly as software advances if not more frequently. Some hardware advances are necessary to run new releases of software. How soon after introduction should these advances in hardware become incorporated into the curriculum? Are students now being exposed to Internet telephony, video conferencing, and live network meetings using chat capabilities? These hardware issues compound the budget constraints especially if a school gets caught with technology advances that do not become part of adopted hardware standards.

The panel discussion will present informed views from administrators, faculty, and industry representatives regarding the approaches that have been used to tackle these issues, what seems to have worked, and what industry expectations are. Participation from the audience will be encouraged.

Visual BASIC Programming: An Introduction

Alka R. Harriger, Professor   Phil Rawles, Assistant Professor

Computer (Information Systems) Technology Department
Knoy Hall of Technology
Purdue University
West Lafayette, IN 47907-1421

Abstract: Applying rapid prototyping to the task of programming is possible through the many "visual" programming languages available today. However, the approach to developing programs using an event-driven "visual" language is quite different than the approach using a more conventional (third generation, non-visual) language. This workshop will show participants how to develop better programs faster by shifting from a conventional programming language to the Microsoft Visual BASIC programming environment. The focus will be on developing general-purpose Visual BASIC applications using a variety of standard and custom features. Participants will share in the joint development of several Visual Basic programs. Upon completion of this workshop, they will receive a diskette with several Visual Basic programs, including those created in the workshop, along with curriculum materials and references (including a list of Internet sites). Collectively, these materials will facilitate their transition to the Visual Basic programming language for their own application development as well as for teaching visual programming.

Workshop Outline

I. Summarize Visual BASIC (VB) usage at Purdue
   A. Introductory programming course
   B. 4-week module in manufacturing networks course
   C. Capstone course with optional application in VB

II. Different types of programming employed in VB
   A. Visual programming: interface development
   B. Event-driven programming: user/system triggers
   C. Structured programming: Coding events

III. Components of VB programming environment

IV. Program development life cycle (PDLC)
   A. Review the approach taught in the classroom
   B. Describe program development tasks at each stage

V. User interface objects
   A. Form: a canvas for screen objects
   B. Controls: a collection of individual screen objects
   C. Properties: attributes of objects

VI. Visual BASIC events: writing structured code
   A. Identify triggers (actions) -> event procedures
   B. Hierarchy chart: tool to show relationships among procedures
   C. Referencing vs. changing object properties
   D. Common event procedures

VII. Visual BASIC Methods

VIII. Programming Standards
   A. Hungarian notation
   B. Option Explicit
   C. GUI Issues

IX. Elementary programming example following PDLC

X. Saving a Visual BASIC program
   A. Multiple files
   B. Copying files to different locations
   C. Creating executable file & distribution package

XI. Printing a Visual BASIC program
   A. Visual BASIC environment
   B. Program executions

XII. Enhancing the program: Adding advanced controls with new events
   A. Add new controls
   B. Add new events
   C. Accessing database data

XIII. More examples demonstrating advanced controls

Building A Reusable Application Framework in Visual Basic

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

Abstract

Of the many benefits of using Objected Oriented Programming (OOP) paradigm to design applications, reusability of software components is probably among the most significant. No one likes to write the same code repeatedly. How many times have you run into situations where you needed a calculation routine that was applicable to more than one application and had to rewrite this routine for each of these applications? How many times have you needed to include your company's standard login screen into a project? Instead of building these common parts of your applications repeatedly, you can create a generic application framework by putting together the most frequently used components in all of your applications. This tutorial will introduce novice participants to a step by step process of building a robust application framework. The tutorial will demonstrate this process by building an application framework, consisting of several reusable components, for a simple case study using Visual Basic. The tutorial will also demonstrate the process for creating an application framework wizard.

Introduction

An application framework may be defined as a functional skeleton forming the basic structure of any application. All additional modules are constructed and fit into this defined framework. Once the application framework is complete, it can be used in any project that requires the functionality offered by the standard components in the generic application framework. The process of incorporating the application framework in a new project can even be automated by creating your own application wizard.

You can therefore empower *all members* of the programming team to conveniently access and include the application framework in any new project. This not only has the likelihood of decreasing the development time but it also tends to improve the consistency of your application development and result in robust applications.

The tutorial will strive to achieve the following objectives:

1. Provide novice participants an understanding of an Application Framework.
2. Introduce the participant to the features in Visual Basic that permit building and deploying an Application Framework.
3. Demonstrate the use of Visual Basic in developing a reusable Application Framework.

The following topical outline is proposed for this tutorial:

1. Introduction to Application Framework
    - 1.1. Define an Application Framework
    - 1.2. Discuss Typical Components of an Application Framework
    - 1.3. Benefits Offered by an Application Framework
2. Process for Designing and Programming an Application Framework
    - 2.1. Identifying components of an Application Framework
    - 2.2. Designing Application Framework Components
        - 2.2.1. Main Module
        - 2.2.2. Splash Screen
        - 2.2.3. Login screen
        - 2.2.4. Main Screen
        - 2.2.5. About Window
        - 2.2.6. Window class
        - 2.2.7. Application Class
3. Visual Basic 4.0 features for building an application framework
    - 3.1. Programming application framework objects
        - 3.1.1. Main Module
        - 3.1.2. Splash Screen
        - 3.1.3. Login Screen,
        - 3.1.4. Window Class,
        - 3.1.5. About Box,
        - 3.1.6. Application Class
    - 3.2. Programming Object Methods & Properties
4. Deploying an Application Framework
    - 4.1. Creating an Application Framework Wizard
        - 4.1.1. Designing an Add-In
        - 4.1.2. Creating an Add-In
        - 4.1.3. Testing the Add-in
    - 4.2. Deploying the Framework Wizard

Use of Cognitive-Based Model for Defining Computer and Information Science Courses

David L. Feinstein, Michael V. Doran, David D. Langan III, Herbert E. Longenecker, Jr.
School of Computer and Information Sciences
University of South Alabama
Mobile, Alabama 36688
(334) 460-6390
doran@cis.usouthal.edu

Workshop Content and Format

The presenters recently completed an NSF-CCD grant that used a cognitive-based approach to develop course materials for the introductory Computer and Information Sciences sequence. This introductory programming sequence is shared by majors in the Computer Science, Information Science, Information Technology, and Computer Engineering degree programs. The topics covered and course content is derived from the relevant curriculum documents for each discipline. The materials produced during the funding period would therefore be of interest to faculty in each of these disciplines. This cognitive-based approach is general and should be of benefit to any course or course sequence.

The workshop will present the key concepts developed during the NSF-CCD grant, the observations of the investigators while using that approach, and will discuss the application of that approach throughout the curriculum. Bloom defined six levels of learning, where each was an extension of the earlier levels. Bloom's taxonomy was used as the basis for developing educational objectives that defined courses in the curriculum. For each topic identified as important to the course, goals and micro-objectives were listed and each of those were mapped to targeted Bloom levels. The proper ordering of the course activities was used to assure that the desired Bloom levels of mastery were reached for each objective.

The use of this approach allows faculty to define courses more explicitly by (a) ordering the behavioral objectives within a course and (b) stating clearly the expectations upon exiting the course. This formal course definition also helps to assure uniformity in the coverage of topics by different faculty members. Additionally, the approach leads to the development of a large body of materials that forms a repository for future instructors to utilize and expand upon.

The format of the workshop will consist of brief presentations by the presenters and several invited participants. The salient features of the approach will be described and the sample materials previously produced will be presented. After these presentations, small working groups will be formed. These working groups will allow attendees of the workshop to apply the approach discussed during the workshop with the guidance of experienced faculty. These working groups will allow participants an opportunity to experience how similar materials might be developed at their home institution. The attendees will achieve level one (fact knowledge) regarding the components of the cognitive-based model. This will be achieved by the presenters reviewing the appropriate educational literature and its vocabulary. The attendees will achieve level two (cued use of the facts) knowledge of the approach by participating with the presenters in the development of materials for a sample topic within a course. The attendees will reach level three (un-cued application of facts) if they (a) use the approach in their own institutions for their own course development, or (b) attend future UFE workshops. In these workshops participants develop modules that are included in the repository at the University of South Alabama.

Background of Presenters

The four presenters for this workshop have: (a) had an NSF-CCD grant funded to explore and develop this cognitive-based approach, (b) successfully used this approach in the classroom throughout the curriculum of CS and IS, (c) had an NSF-UFE grant funded to present these concepts in a week-long workshop, (d) published numerous papers regarding this approach, and (e) been involved in national IS and CS curriculum efforts.

The presenters will be joined by several previous participants from summer 1996 UFE workshop. These prior participants include: Marion Harmon (Florida A&M), Art Jones (Morehouse College), Ken Messa (Loyola of New Orleans), D. C. White (U of South Florida), Irv Englander (Bentley College) and L. Wayne Horn (Pensacola Jr. College). These members of the on-going working group will briefly present their observations and report on their use of the cognitive-based approach. They will also act as guides and assist workshop participants in the use of the approach when small groups are formed during this ISECON workshop.

Publishing on the Internet

Dr. Wayne C. Summers
Computer Science Department
New Mexico Highlands University
Las Vegas, New Mexico 87701
E-mail: wsummers@venus.nmhu.edu

This tutorial will focus on developing a web site. The first part of the tutorial will discuss the design and creation of web documents while the second part will look at the actual publishing of the documents on the web.

Part I of the tutorial will begin by looking at good and bad home pages on the Web. We will look at examples of what works and what doesn't work. A variety of design issues and strategies will be discussed. Home pages are written in HTML - HyperText Markup Language. The language is written in plain text and can be written using any editor or word processor and saved as an ASCII file. To enhance the text and provide links to other documents and graphics, the developer includes tags that identify the way the text will look and directions for branching to other documents. These tags can be entered along with the text or can be inserted automatically if the developer uses HTML design software. A brief overview of the different types of tags will be given as well as how the tags are used. HTML documents can now be created with most word processors as well as a wide variety of specialized editors for creating these documents. The advantages and disadvantages of using these tools will be discussed and illustrated through examples.

Web pages have become more dynamic in the last year through the use of advanced HTML features and enhancements provided by using programming languages and environments like CGI, Java, JavaScript and ActiveX. These enhancements will be compared and discussed. Examples created using these tools will be explained as well as illustrations of how to incorporate them into web pages. This section of the tutorial will conclude by looking at the latest developments in HTML and web page design and development.

Designing and writing a web page is only half of publishing on the Web. The second part of this tutorial will include the installation of a web server and a discussion of what is involved in maintaining the web site. There are several web servers readily available for all platforms. Installation of a web server is not difficult as will be illustrated during this tutorial. A few of these web servers for Windows '95 and Windows NT will be discussed and compared. Problems in setting up a web site and maintaining it will also be considered.

The workshop will conclude with a discussion of how developing home pages can be used in the classroom. The discussion will include a look at how educators are incorporating the web into their classes.

All of the materials and examples for this tutorial will be accessible from the author's web site at http://jaring.nmhu.edu for use by the participants.

USING LEARNING STYLES to PROMOTE ACTIVE LEARNING in a COMPUTER LITERACY COURSE

Joseph H. Daniel, Assistant Professor
Delaware Valley College
700 East Butler Avenue, Doylestown, PA 18901
danielj@devalcol.edu (215-489-2498)

Barbara F. Daniel, Director of Student Affairs
Penn State University, Delaware County Campus
25 Yearsley Mill Road, Media, PA 19063
bfd1@psu.edu (610-892-1457)

ABSTRACT

A multiple regression model proves that a "thinking learning style", a "doing learning style", and "total college credits" are the best variables that describe the "A" students in a computer literacy course. With these students as "expert learners" in a student team and the application of teaching techniques for active learning, the faculty in this literacy course can shift their teaching style from the passive, lecture-only approach to one of active participation by students.

INTRODUCTION

Results from a multiple regression model clearly indicate a need for faculty to shift their teaching style from the passive, lecture-only approach to one of active learning by students. Even though the lecture still remains the primary technique of providing background information, concepts, and methods for the student, the student can build on this foundation to become a more active learner in the classroom.

A regression model was developed specifically to determine what learning styles are used by students who earn "A" grades in a computer literacy course which uses the Microsoft OFFICE software. This model proves that "thinking" and "doing" learning styles are among the best variables that describe these "A" students.

LEARNING VARIABLES

The learning styles of "thinking" and "doing", with their opposites of "feeling" and "watching", are measured by a Learning-Style Inventory of twelve questions, which is based on David Kolb's model of experiential learning and produced by the McBer and Company. This inventory quantifies each learning style on a percentile scale that is based on the scores of 1,446 adults ranging from 18 to 60 years of age.

The Learning-Style Inventory recently was taken by 167 college students in a computer literacy course where 55 of the students eventually earned an "A" grade in the course. Their grade was based on twelve laboratory assignments and six examinations in the word processing, spreadsheet, graphics, database, and integrated aspects of the Microsoft OFFICE software package.

The data from the Learning-Style Inventory and course grade distribution are placed into a multiple regression model to explain the linear relationship between a single dependent variable for the "course grade" and several independent variables, such as the "thinking percentile", "doing percentile", "feeling percentile", and "watching percentile" from this inventory.

Other independent variables considered include the student's gender, college major, college minor, hours worked on this course per week, willingness to learn about computer topics, semester level in college, total college credits taken before this course, and, years of computer programming experience in industry.

Feeling	Learn by Concrete Experience (CE); Relate to people in everyday problem
Watching	Learn by Reflective Observation (RO); Understand problem from different views
Thinking	Learn by Abstract Conceptualization (AC); Use logic and ideas to understand problem
Doing	Learn by Active Experimentation (AE); influence people and events through action

FIGURE #1 - Learning Styles (Kolb 85)

REGRESSION MODEL

The overall strength of the relationship between these variables is measured by the coefficient of multiple determination (or R2). The R2 for this model showed that 90% of the variation in the "course grade" of "A" students is explained by the variation in the independent variables of "thinking percentile", "doing percentile", and "total college credits".

Since R2 does not accept or reject a hypothesized relationship between variables, the interpretation of the partial regression coefficients is the next step. Each regression coefficient is interpreted as one unit change in which the dependent variable produces a change in the independent variable which is equal to the value of its regression coefficient.

Thus, the regression equation contains a 86.9 constant, 0.03 regression coefficient for the "thinking percentile" variable, 0.06 regression coefficient for the "doing percentile" variable, 0.03 regression coefficient for the "college credits" variable, and a 0.37 error term. An error term reveals the fact that the data points do not neatly fall on this regression line.

Dependent	"Course Grade" is changed or predicted by one or more independent variables
Independent #1	One unit change in "Thinking Percentile" causes 0.03 change in "Course Grade"
Independent #2	One unit change in "Doing Percentile" causes 0.06 change in "Course Grade"
Independent #3	One unit change in "College Credits" causes 0.03 change in "Course Grade"

FIGURE #2 - Regression Equation (Wonnacott 90)

DATA ANALYSIS

The statistics from this regression model show that success in a computer literacy course is most likely accomplished by the "A" students who apply "thinking" and "doing" learning styles. (See the Statistical Notes at the end of the paper) Early identification of these students through a Learning Style Inventory can help faculty create a classroom learning environment which benefits all the students in the course.

This environment should include elements of active learning that center on "thinking" and "doing" as well as learning partnership that pair the "A" student as an "expert learner" with one or two "weaker" students in a team. The learning activities of this partnership should involve the students in talking, writing, reading, and reflecting on the concepts presented in the course lectures.

Talking clarifies thinking and is done, for example, when a team of two or three students discusses lecture concepts and performs laboratory assignments together. Teams also provide an opportunity for the students to learn important interpersonal skills.

Like talking, writing helps the student to become a clearer thinker. A research paper in a computer literacy course, for example, helps students to explore their thinking through the organization and clarification of concepts. Research-oriented topics, such as the Internet, captivate the interest of students and encourage their active participation.

Reading requires the student to understand what others think, as opposed to clarifying the student's own thoughts by talking and writing. The faculty member should provide some structure to the student's reading efforts, for example, by handing out an outline of the chapter for the current week's lecture.

Reflecting allows the student time to think about concepts presented in the lecture. One such technique for reflecting involves a student journal which requires periodic entries on what the student thinks and feels about issues, concepts, and events in the week's lecture and laboratory.

Talking	Clarifies thinking by doing laboratory assignments in a team of students
Writing	Explores student's own thinking about concepts when doing a research paper
Reading	Understand what others think by doing a weekly text book reading assignment
Reflecting	Comments on what student thinks about issues and concepts by doing a journal

FIGURE #3 - Active Learning (Meyers 93)

TEAM LEARNING

In order to promote learning partnerships, the potential "A" student (identified by the Learning Styles Inventory) is teamed with one or two other students to perform laboratory assignments (42% weight). The remainder of the course is based on individual effort. Teams are periodically reformed so that students get experience with more students in the class.

How well a team operates depends on the clarity of its objectives (why), parameters of its activity (what), and guidelines agreed upon for interaction (how). Suggested behavior guidelines for the small groups include (1) listen when one student is talking, (2) respect everyone's contribution, (3) arrive at consensus through talking, and (4) do not dominate the discussion.

Recorder	Records ideas from team's discussion
Encourager	Involves everyone in team's discussion
Clarifier	Understands what everyone is proposing
Mediator	Helps members to arrive at consensus
Timekeeper	Gives members equal time in discussion

FIGURE #4 - Team Roles (Meyers 93)

DATA SYNTHESIS

Separately, talking, writing, reading, and reflecting, each in their own way, involve a different type of "thinking" and "doing", but together these techniques help in the active learning of the student. These techniques and the learning partnerships recently have been incorporated into the revised structure of a college computer literacy course.

For example, the syllabus given to the students explains that the final grade for the computer literacy course is now based on six laboratory assignments (42% weight), two examinations (36% weight), one research paper (14% weight), and one journal (8% weight) on the word processing, spreadsheet, graphic, and presentation aspects of Microsoft OFFICE.

The weekly lecture topics on this syllabus comprise: (1) operating system, (2) computer system, (3) word processing, (4) secondary storage, (5) central processor, (6) input/output, (7) spreadsheet, (8) database management, (9) computer program, (10) information system, (11) communications, and (12) ethical issues.

What the student thinks about these lecture topics are recorded in a journal, which is not for lecture notes, but for reflecting and writing on the observations, concerns, and frustrations by the student. A rating scale of "good insights", "adequate", and "inadequate" is used to evaluate the journal.

STUDENT VARIABLE

The third independent variable in the multiple regression model was "total college credits" that were taken by a student before the computer literacy course. If the number of credits is high, then the probability of earning an "A" grade is high. This could be interpreted as a student who has more experience with college courses is more likely to earn higher grades.

The application of any classroom theory requires feedback from the students in order to determine the effect the method is having on the student. This is the purpose of a survey which is administered every four weeks in the computer literacy course to ask the student (1) What do you like best about the course?, (2) What could the course be doing even better?, and (3) Explain ways in which the course is not meeting your expectations?

This monthly survey also promotes active learning since it establishes the students as partners in the learning process throughout the course. Their input constantly refines classroom activities so that the needs of all students are met. Daily interaction with the students by asking them directly how things are going will provide faculty members with information and insights into what and how well their students are learning in this active classroom environment.

STUDENT LEARNING

According to David Kolb, the "thinking" style involves the analysis or breaking down of a problem into its understandable components. These components are then reassembled into a logical plan for solving the problem, for example, by means of a Data Flow Diagram. The "converger" (or Systems Analyst) relies on systematic planning to solve technical problems.

In addition, the "doing" style involves the implementation of this logical plan, for example, by coding a Structure Chart into a computer program. The "converger" may also influence people and events in the development of a program. Once it is completed, the "converger" (or Programmer) takes great pride in seeing the results of using the computer to solve the problem.

Diverger	Combines CE and RO; Compares objective with reality and identifies problems
Assimilator	Combines RO and AC; Selects problem and considers alternative solutions
Converger	Combines AC and AE; Evaluates solution consequences and chooses best solution
Accommodator	Combines AE and CE; Implements solution and then finds a new model or objective

FIGURE #5 - Learning Cycle (Kolb 85)

CONCLUSION

By finding practical uses for ideas and theories, dealing with technical tasks and problems, and making decisions based on finding solutions to these problems, a student, according to David Kolb, is reinforcing the "converger" learning style. A "converger" learns primarily by combining the "thinking" and "doing" activities in solving technical problems.

With these students as "expert learners" in a student team and the application of teaching techniques for active learning, the faculty in a computer literacy course can successfully shift their teaching style from the passive, lecture-only approach to one of active participation by students.

STATISTICAL NOTES

The statistical significance of a partial regression coefficient is measured by the t-statistic (T) which is the ratio of the coefficient to its standard error. Since the calculated T of 4.2 for "thinking percentile", 8.3 for "doing percentile", and 7.7 for "college credits" are all greater than the published T of 2.3, then each coefficient is accepted at a 95% level of confidence.

Another important test of significance for this regression model is the F-statistic (F) which is the ratio of the explained variance divided by the unexplained variance. Since the calculated F of 30.2 is greater than the published F of 4.4, then all of the partial regression coefficients as a complete equation are accepted at a 95% level of confidence.

One of the assumptions underlying regression analysis is that there is no heteroscedasticity or no relationship between the size of the independent variables and the residual, which is the difference between the observed and predicted values of the dependent variable. Plots of the residual against an independent variable showed no pattern, so no homoscedasticity exists.

Another assumption underlying regression analysis is that there is no multicollinearity or no relationship between the independent variables. A correlation matrix revealed that all correlation coefficients are less than 0.6, so no linear relationship exits among the independent variables of "thinking percentile", "doing percentile", and "total college credits".

BIBLIOGRAPHY

Boyer, Glen, and McKell, Lynn. "Information Systems Knowledge and Skills for the Business Management Undergraduate." Journal of Information Systems Education, Spring, 1996, pp.21-28.

Cornwell, John. "Kolb's Learning Style Theory Revisited." Educational and Psychological Measurement, Summer, 1995, pp.317-327.

Dunn, Rita. How to Implement and Supervise a Learning Style Program. Alexandria, VA: Association for Supervision and Curriculum Development 1996.

Enerson, Diane, and Plank, Kathryn. The Penn State Teacher. University Park, PA: Penn State University, 1993.

Kolb, David. Learning Style Inventory. Boston, MS: McBer and Company, 1985.

Kyriacou, Chris, and others. "Pupil Learning Styles and Foreign Language Learning." Language Learning Journal, March, 1996, pp.22-24.

Lynott, Donna. "Teachers' Implicit Theories of Intelligence." Journal of Research and Development in Education, Summer, 1994, pp.253-264.

Martin, Edward, and Parker, Charles. Mastering Today's Software for Windows 95. Orlando, FL: Dryden Press, 1995.

McCarthy, Bernice. Learning Styles and Schools: Making It Happen. Chicago, IL: 1991.

Myers, Chet, and Jones, Thomas. Promoting Active Learning. San Francisco, CA: Josey-Bass Publishers, 1993.

Norusis, Marija. SPSS/PC+ Advanced Statistics. Chicago, IL: SPSS Inc., 1988.

Parker, Charles. Understanding Computers. Orlando, FL: Dryden Press, 1996.

Schroeder, Charles. "New Students - New Learning Styles." Change, September-October, 1993, pp.21-26.

Wonnacott, Thomas, and Wonnacott, Ronald. Statistics for Business and Economics. New York, NY, CA: Wiley, 1990.

Skill Requirements for Entry-Level IS Graduates: A Report from Industry

Craig Van Slyke
ISDS
University of South Florida
4202 E. Fowler Ave.
CIS 1040
Tampa, FL 33620-7800
(813) 974-6767
cvanslyk@bsn01.bsn.usf.edu

Marcy Kittner
CIS
The University of Tampa
401 W. Kennedy Blvd.
Tampa, FL 33606
(813) 253-3333
mkittner@alpha.utampa.edu

Paul Cheney
ISDS
University of South Florida
4202 E. Fowler Ave.
CIS 1040
Tampa, FL 33620-7800
(813) 974-5524
pcheney@bsn01.bsn.usf.edu

Abstract

Determining what skills employers of new IS graduates desire is important in guiding IS educators in designing curriculum and in advising students. This study reports the results of a pilot study which attempts to gain insight into the previously mentioned question. The findings indicate that basic technical skills, such as systems analysis and design, and database concepts, and non-technical skills, such as teamwork and communications skills, are more important than specific technical skills.

Introduction

There is much debate over how universities and colleges should prepare their graduates for successful entry into the job market. Information Systems (IS) departments are faced with the same issues that drive the aforementioned debate: how to educate their students so they can get good jobs, and also be successful in their careers. In addition, IS departments are faced with problems related to the dynamic nature of information technology (IT). Issues such as the emergence (and possible decline) of client/server and the associated decline in mainframes, how to harness the power of the Internet, and the rise in end-user computing all raise questions regarding how IS educators should prepare their students for the "new" world. Which language(s) should be emphasized--C, COBOL, Visual Basic? Should Data Communications be a required course? These and other questions require a knowledge of the nature of the IS profession, including how to successfully enter that profession.

While several studies in the recent IS literature have examined what skills help IS professionals advance in their careers (Cheney, et al., 1990; Leitheiser, 1993; Lee, et al, 1995), few have studied the skills that employers look for when hiring recent graduates. This is the purpose of the study reported here--to try to determine just what skills employers find important when making hiring decisions regarding recent IS graduates. This paper reports the results from the first phase of an investigation into the importance placed on various skills.

The paper is organized as follows. We begin by setting the scope of the project and discussing why it is important. This is followed by a discussion of previous research that relates to this project. Next, the methodology used in this study is described. Then we present the study's results--including how they compare with the recent IS'95 curriculum guidelines (Cougar, 1995). Finally, we discuss the study's implications and draw some conclusions.

Scope and Importance of the Study

The study reported here is preliminary in nature--it is the first phase of a two phase project. This phase has two goals. First, we hope to gain preliminary insight into the following research question: *What skills do employers of recent IS program graduates find important when evaluating job candidates for entry-level positions?*

Also, as an extended pilot study, we want to refine our data gathering instrument primarily by refining the list of skills. While the results from any preliminary study should be taken with some skepticism, we feel that the sample (described in the Methodology section) represents a cross section of IS employers and that the results reported here may be at least somewhat representative of the feelings of IS employers as a whole.

It is important for IS educators to understand what employers desire when searching for and selecting new graduates for career positions (Nelson, 1988). Research such as that reported here can help educators in several ways. First, the results can help guide curriculum decisions (Nelson, 1988), such as what courses to require, or what languages to teach. In addition, individual educators can use the results to help them in designing individual courses. For example, the heavy importance placed on team and group work (see the Results section) may lead educators to emphasize team projects more than they would otherwise. Finally, the results can be used to help with advising duties. Students routinely seek guidance regarding course selection, and a good knowledge of what employers seek in new graduates can clearly help educators provide students with sound advice.

Previous Research

IS researchers have paid little attention to entry-level job skill requirements. Studies have examined skill requirements within specific IS job classifications, such as systems analysts (Misic, 1996), as well as skill requirements for IS professionals in general (Lee, et al., 1995; Athey, et al., 1995-1996; Prabhakar, 1996), but have not specifically investigated skill requirements for entry into the IS profession. In this section we briefly discuss the findings of previous studies which have research job skill requirements.

Successful IS professionals need to possess both good technical skills and interpersonal and communications skills (Nelson, 1988; Cheney, et al., 1990; Leitheiser, 1993). Most extensive studies of skill requirements find that the "soft" skills - human relations, communications, organizational skills and the like - are at least as important as technical skills, especially as the professional moves into more managerial roles (Nelson, 1988; Cheney, et al., 1990; Lee et al., 1995).

The practitioner literature has also examined IS skill requirements. Studies of skill requirements for systems analysts tend to agree with the findings of academic researchers--a mix of interpersonal, technical and organizational skills provides for a sound skill foundation for the IS professional (Blank & Barratt, 1988; Misic, 1996).

As we have seen in this section, there is some level of agreement as to what skills are desirable for IS professional to possess. However, none of the studies cited specifically examines the skill set necessary for successful entry into the IS profession, which is the subject of this study.

In the following section, the methodology used to investigate the previously mentioned research question is described.

Methodology

In this section, we discuss the methodology used in this study. We begin by describing the instrument development process, follow with a description of the sample used, and conclude by describing how the instrument was administered.

The first step in developing the instrument was to develop an initial list of skills to include in the questionnaire. In order to accomplish this we drew from several sources, including recruiting material from major employers, employment listings obtained from university placement departments, internship listings, help wanted advertisements and informal discussions with placement personnel. Skills included in prior studies were also included. Both authors performed this step independently, and the lists were then consolidated. This consolidated list was aggregated into groups of related skills, including general IS skills, programming languages, operating systems, networking, database management systems, other IS skills, general skills and business-related skills. The authors discussed and reached agreement on the groupings.

After constructing the initial skills list, a draft of the actual instrument was prepared and administered to several knowledgeable individuals, including industry professionals and faculty members with industry experience and/or knowledge of the IS job market. This pilot test served several purposes. First, comments from the pilot test were used to improve the skills list and the questionnaire layout. Subjects for the pilot test were also asked to mark how long they spent in taking the survey. This estimate was included in a participation request message sent to possible subjects.

In order to get a reasonably representative cross-section of employers for this survey we used several sources to obtain possible participants. These sources included local IS-related help wanted advertisements, recruiting advertisements in *Computerworld's Campus Edition* (1996), job posting on several Internet employment sites, and national employers with local offices known to the authors. An email message or phone call was used to request the organization's participation which resulted in 35 favorable responses.

The sample includes organizations from a variety of industries, including utilities, consulting, government, financial services, insurance, education, software development, and retail. The respondents are also geographically dispersed, although the majority are from the central Florida area. While this sample certainly can not be considered to be random, it does seem to represent a cross-section of employers and may be satisfactory for a preliminary study.

The questionnaire was administered both in paper form and via email. Although the sample size in this preliminary study is insufficient to test for significant differences between the paper and email questionnaire, visual inspection did not reveal any consistent differences. Subjects were asked to rate each skill on a scale of 1 (not important) to 5 (critical). Respondents also had the opportunity to add skills to the given list.

Results

Analysis of the questionnaires received to date reveals some interesting information, which is reported in this section. Each group of skills will first be discussed individually. This is followed by a brief comparison of these results to findings of earlier studies, and we finish by drawing some general conclusions regarding the results. Summary results are shown in Appendix A. Detailed results appear in Appendix B.

General IS Skills: The responses for this skill group display variability in the mean ratings for the various individual skills, although not as much as some of the other skill groups. While the group mean rating was above the overall mean rating (3.582 vs. 3.414), some of the skills were rated relatively low. For example, CASE tools (2.400) and telecommunications (2.933) were both below the overall mean. Conversely, database concepts (4.533), systems analysis and design (4.267) and structured programming (4.063) were among the more highly rated skills. Informal discussions with recent graduates indicates that many employers are "quizzing" job prospects on their systems analysis and database concepts skills by asking applicants to read data and process models and by asking extensive questions on database concepts.

There were some surprising results in this category, most notably the low rating of telecommunications. Given the recent interest in electronic commerce, the Internet, and intranets, the authors anticipated that telecommunications would be more highly rated. There are several possible explanations for this result. It may be that employers look more toward computer science graduates for their telecommunications needs. Another possibility is that these subjects were more interested in specific areas within the telecommunications umbrella, especially given the high rating of the networking skill group (group mean of 3.700, discussed later). Another possibility is that the term "telecommunications" is simply too general. It should be noted that the relatively high standard deviation (1.387) in the ratings indicates that our respondents were not in agreement as to the importance of telecommunications skills.

Programming Languages: The skills in the programming languages group received, in general, relatively low ratings. The mean rating for the skills in the group (2.862) was well below the overall mean (3.414). However, there was significant variability in the ratings within the group as evidenced by the group standard deviation of 1.441. Not surprisingly, several "hot" programming languages were rated highly such as Visual Basic (4.000), C (3.647), and C++ (3.706).

Not surprisingly, "older" languages did not fare well. ADA (1.733), RPG (1.667) and Assembler (1.733) all received ratings well below the group and overall means. The authors were somewhat surprised by the relatively low rating of COBOL (2.733), given recent reports of signing bonuses for COBOL knowledgeable graduates. It may be that employers feel that they can train individuals in programming COBOL as long as they are equipped with good basic skills. This is especially true of large organizations--traditionally heavier users of COBOL--since they have the resources to train new graduates. Anecdotal evidence from some of our recent graduates indicates that this may be the case, although no firm conclusions should be drawn. It is also interesting to note the relatively high standard deviation of the COBOL ratings, indicating that our subjects were not in agreement as to the importance of COBOL skills.

Operating Systems: Similar to the programming skills group, the operating systems (OS) skill group shows wide variability in the ratings of individual skills with a group standard deviation of 1.498. In general, PC and workstation OS's tend to be rated highly, while mainframe OS's were rated well below the overall and group (3.094) means. The exception to this generalization is with O/S 2, which had a very low rating of 1.933. Our respondents rated skill with Windows/Windows 95 (4.375) and Unix (4.294) as important, while JCL (2.375), VMS (1.933) and CICS (2.00) were rated as being relatively unimportant.

The ratings in this category were not unexpected. Even though the predictions of the disappearance of the mainframe were premature, it seems reasonable that there is an existing supply of professionals knowledgeable in mainframe OS's, so employers look to recent graduates for knowledge in newer operating environment.

Networking: As a skill group, networking was rated higher than the overall mean (3.700 vs. 3.414). All but three of the skills in this category (email administration, system security, and interoperability) received ratings above the overall mean. However, unlike the programming category, none of the low rated skills were substantially below the overall mean. It appears that almost any networking skill is desirable, but particularly popular were Windows NT (4.313), Novell Netware (4.125) and client/server (4.294) skills. This is not surprising considering the current popularity of these areas. It is also interesting to note that in the higher rated skills, the standard deviation of ratings was low relative to the lower rated skills. It appears that there is general consensus on the "hot" skills--a consensus not reached on the less popular skills.

Database Management Systems: As in the programming languages and operating systems categories, the database management system (DBMS) skill group exhibits wide variation in how subjects rated the various skills (standard deviation of 1.420). Although the group mean of 3.278 is slightly below the overall mean (3.414), several of the individual skills received ratings well above the mean. For example, Oracle (3.938), and Sybase (3.600) , both proprietary DBMS, and SQL (4.375), a nonproprietary database query language, were all rated above the overall mean. This is not an unanticipated result--Oracle, Sybase and SQL are increasingly popular in industry. Informal discussions with recent graduates confirm that many employers are indeed interested in database skills in general, and in Oracle and SQL skills in particular.

Other Information Systems Skills: The skills in this grouping were placed in this category only because there was not a natural place for them in another group. Because of this, it seems inappropriate to discuss the group as a whole, so the discussion is restricted to individual skills within the group. Only two of the skills in this group, PC applications (3.563) and end-user support (3.438) were rated higher than the overall mean rating (3.414). This may reflect the fact that these skills are more general in nature, while the lower rated skills in this category tended to be more specialized.

General Skills: This skill group consisted of skills which were not technical in nature, and did not fit into the business-related skills group. These skills were seen by respondents as being of considerable importance, as reflected by the mean rating of the skills in this group (4.373), which is much higher than any other skill group. The networking group, the second most highly rated group, had a mean rating of over 0.60 points lower than the this group's. There was also little variability in the ratings. The overall standard deviation for the group (0.709) was much lower than the group standard deviations of any other group, indicating that these subjects were in general agreement on the ratings. All of the skills in this group received ratings higher than the overall mean (3.414). In fact, the lowest rated skill in this group, leadership (3.706) was still considerably higher than the overall mean. The highest rated skill in this group, analytical ability (4.824), was also the most highly rated overall. Both oral (4.529) and written (4.353) communications skills were highly rated, as was listening skill (4.529), confirming what many IS educators have been telling students for some time. It also appears that the ability to work in groups and teams is important to employers, as evidenced by the high ratings for working with others (4.471), teamwork (4.529), and ability to work in groups (4.529). Employers also seem to desire IS employees who are self-motivated (4.647).

The message sent by these subjects seems consistent with that sent by the practitioner press--"soft" skills are increasingly important for IS professionals. Several recent articles indicate that executives are demanding that IS professionals become more skilled in such areas as writing memos and listening closely to their associates (Davis, 1993) and that communications and interpersonal skills are more important than technical skills (Misic, 1996).

Business Related Skills: The authors were somewhat surprised by the mediocre ratings of the skills in the business related skills group (group mean of 3.402). Only understanding business function areas (3.882) was rated well above the mean, although user interviewing (3.471) and quality assurance (3.471) were both slightly above the overall mean (3.414). We expected that the skills in this category would be more highly rated. It is possible that employers feel that if new hires possess the types of skills found in the general skill group they can acquire the business related skills once they are on the job.

The findings presented here seem to be in general agreement with prior studies of the skill requirements of IS professionals. This study's findings on the importance of interpersonal, communications and analytical skills agree with findings of earlier studies which point to the need for interpersonal, communication and business skills for experienced IS personnel (Leitheiser, 1993; Lee et al., 1995).

In this section, we presented and discussed the results of our survey. In some cases, the results followed what most IS educators might suspect; in others, the results were quite surprising. But the employers surveyed in this study seem to be sending some loud and clear messages. First, the importance of general thinking, communication, and interpersonal skills such as analytical, teamwork, and listening skills is clear, at least in this sample. Employers want IS graduates who can think, communicate, and work well with others. They also want individuals who have a good grounding in basic IS skills such as systems analysis and design and database concepts. These employers seem to be less interested in more specific technology skills.

In the next section, we will examine how well our results match with the often cited IS'95 curriculum guidelines (Cougar, et al., 1995).

Comparison to IS'95 Curriculum Guide

One of the possible implications of this research is its impact on curriculum design. Skills assessments and curriculum guidelines are both important in helping IS departments prepare for jobs in IS. In that vein, a comparison of the findings of this survey to the IS'95 undergraduate curriculum guideline's attributes of IS program graduates (Cougar, et al., 1995) may be interesting. IS'95 is a report from a joint curriculum development effort of the Association for Computing Machinery, the Association for Information Systems, the Data Processing Management Association and the International Conference on Information Systems. In addition to providing a model undergraduate IS curriculum, IS'95 describes output attributes of graduates. In an effort to compare our results to the desired attributes as expressed in IS'95, we mapped skills from our survey onto the IS'95 graduate attributes. (A summary of this mapping is available from the first author.)

For the most part, the skills that were rated highly by our respondents fit with the desired attributes of IS graduates as expressed by IS'95. (The interested reader is directed to the published IS'95 guideline for further explanation of the attribute categories.) For example, communication skills received universally high ratings by our respondents and are also noted as desirable attributes of IS graduates by IS'95. The same holds true for the other attribute categories.

Conclusions

Based on the preliminary results of this study, it seems that employers find basic technical, and "soft" skills more important than more specialized technical skills when evaluating recent IS graduates for entry-level positions. This is, in general, in agreement with previous studies of the skill required as IS professionals advance in their careers. Further, it appears that these results are also, generally, in keeping with the recent IS'95 curriculum guidelines (Cougar, et al., 1995).

References

Athey, S. and Wickham, M. (1995-1996). Required skills for information systems jobs in Australia, *Journal of Computer Information Systems*, 36(2), 60-63.

Blank, M. and Barratt, D. (1988). Finding and selecting systems analysts and designers, *Journal of Systems Management*, 39(3), 8-11

Cheney, P., Hale, D. and Kasper, G. (1990). Knowledge, skills and abilities of information systems professionals: Past, present and future, *Information & Management*, 19(4), 237-247.

Cougar, J. Davis, G., Dologite, D., Feinstein, D., Gorgone, J., Jenkins, A., Kasper, G., Little, J., Longenecker, H., and Valacich, J. (1995). IS'95: Guideline for undergraduate IS curriculum, *MIS Quarterly*, 22(3), 341-359.

Dwight, D. (1993). Hard demand for soft skills, *Datamation*, 39(2), 28-32.

Lee, D., Trauth, E. and Farwekk, D. (1995) Critical skills and knowledge requirements of IS professionals: A joint academic/industry investigation, *MIS Quarterly*, 19(3), 313-340.

Leitheiser, R. (1992). MIS skills for the 1990's: A survey of MIS managers perceptions, *Journal of Management Information Systems*, 9(1), 69-91.

Misic, M. (1996). The skills needed by today's system analysts, *Journal of Systems Management*, 47(3),34-40

Nelson, R. (1988). Educational needs as perceived by IS and end-user personnel: A survey of knowledge and skill requirements, *MIS Quarterly*, 15(4), 502-525.

Prabhakar, B., Litecky, C. and Arnett, K. (1996). A longitudinal analysis of job skill trends in the MIS job market, *Proceedings of the Americas Conference on Information Systems*, August 16-18, Phoenix, 338-340.

Appendix A - Summary Results

Skill Category	Mean	Median	St. Dev.
General IS	3.582	4.0	1.179
Programming Languages	2.862	3.0	1.441
Operating Systems	3.094	3.0	1.498
Networking	3.700	4.0	1.097
DBMS	3.278	4.0	1.420
Other IS	3.024	3.0	1.297
General	4.373	4.0	0.709
Business Related	3.402	3.0	1.171

Appendix B - Detailed Results

Skill	Mean	Median	St. Dev.
General IS			
Sys. An. & Des.	4.267	5.0	0.961
CASE tools	2.400	2.0	0.986
Database concepts	4.533	5.0	0.640
System testing	3.938	4.0	0.929
DSS an. & des.	3.200	3.0	1.265
Telecomm.	2.933	3.0	1.387
OO an. & des.	3.467	4.0	1.125
Prototyping	3.333	4.0	1.113
Documentation	3.625	4.0	1.025
Structured prog.	4.063	4.0	0.772

Skill	Mean	Median	St. Dev.
Prog. Lang.			
COBOL	2.733	3.0	1.668
C	3.647	4.0	1.412
C++	3.706	4.0	1.404
Visual Basic	4.000	4.0	1.155
4GL's	3.071	3.0	1.328
HTML	3.353	4.0	1.115
RPG	1.667	1.0	0.976
ADA	1.733	1.0	1.163
Assembler	1.733	1.0	0.961
Unix Shell Script	3.000	3.0	1.211
Java	3.063	4.0	1.237
Delphi	2.267	3.0	1.163
Operating Systems			
Unix	4.294	4.0	0.772
MS-DOS	3.563	3.5	1.263
Windows / 95	4.375	5.0	0.806
OS/2	1.933	2.0	0.884
Windows NT	3.941	4.0	0.899
JCL	2.375	2.0	1.455
VMS	1.933	1.0	1.335
CICS	2.000	1.0	1.363
Networking			
Windows NT Server	4.313	4.0	0.602
Novell Netware	4.125	4.0	0.619
Client/Server	4.294	4.0	0.686
Email administration	2.938	3.0	1.289
System security	3.133	3.0	1.060
TCP/IP	3.875	4.0	1.088
Interoperability	3.071	3.0	1.207
DBMS			
Clipper	1.733	1.0	0.961
DB2	2.667	3.0	1.447
Informix	3.133	3.0	1.356
MS Access	3.375	4.0	1.455
Oracle	3.938	4.0	1.289
SQL	4.375	4.5	0.719
Sybase	3.600	4.0	0.986
"Other" IS			
Distributed systems	3.133	3.0	1.246
EDI	3.200	3.0	1.265
MS Back Office	2.867	3.0	1.506
PC applications	3.563	4.0	1.315
Lotus Notes	2.875	3.0	1.360
AI development	2.067	2.0	1.033
End-user support	3.438	4.0	1.209
End-user training	3.000	3.0	1.095

Skill	Mean	Median	St. Dev.
General			
Analytical ability	4.824	5.0	0.393
Assertiveness	3.882	4.0	0.781
Creativity	4.235	4.0	0.664
Listening	4.529	5.0	0.514
Oral communications	4.529	5.0	0.624
Written communications	4.353	4.0	0.702
Organizational ability	4.294	4.0	0.772
Prioritizing tasks	4.353	4.0	0.702
Self motivation	4.647	5.0	0.606
Working with others	4.471	5.0	0.624
Teamwork	4.529	5.0	0.624
Ability to work in groups	4.529	5.0	0.514
Problem solving	4.647	5.0	0.493
Critical thinking	4.294	4.0	0.686
Problem diagnosis	4.353	4.0	0.493
Ability to follow directions	4.353	4.0	0.702
Ability to work with users	4.176	5.0	1.015
Leadership	3.706	4.0	0.985
Business Related			
Understand business function areas	3.882	4.0	0.993
Perform cost/benefit analysis	3.294	3.0	1.312
Perform feasibility analysis	3.059	3.0	1.435
User interviewing	3.471	3.0	1.231
Project management	3.235	3.0	1.091
Quality assurance	3.471	3.0	0.874

Beyond the Classroom: Mentoring in the CIS Academic Community

Michael V. Doran, R. J. Daigle, Robert A. Robertson
School of Computer and Information Sciences
University of South Alabama
Mobile, Alabama
doran@cis.usouthal.edu

Abstract

Elliot Soloway, noted author and teacher, recently observed in a presentation at ACM'97 in San Jose, that human interaction and nurturing are as much a part of the educational process as is the discipline knowledge. He went on to state that no use of technology can replace this human element. Although the classroom is the traditional instructional forum for issues such as professional ethics, responsibility to society, and the need for a life-long learning, a more individualized approach to learning is recognized as providing a higher degree of success [Parks 1990, Parks 1995]. How can educators provide a more individualized approach to learning without sacrificing classroom content? One answer is *Mentoring*. Several years ago, our university established a mentoring program for certain high-risk students. The School of Computer and Information Science (CIS) embraced the mentoring concept and extended it to involve CIS majors for in-class and out-of-class activities. In this paper we describe the mentoring process as it occurs in our CIS community.

Introduction

Many CIS curriculum documents [ACM 1991, DPMA 1991] outline a holistic approach to learning which results in a complete student. Graduates of CIS programs are expected to achieve a personal as well as a professional development. They are expected to possess sufficient professional skills and knowledge to gain initial employment, and they are expected to possess sufficient personal skills to work effectively in teams and to be productive members of society. They are expected to have a sense of personal as well as professional ethics and to understand their individual role as well as professional role in society. They are viewed by members of society as possessing valuable technical skills and problem solving abilities and, therefore, leadership roles gravitate towards them. How can we as faculty give more than curriculum instruction so that our students are prepared for these leadership roles? How can we get them to commit to a life-long learning approach to cope with new professional and community demands?

Academic environments are also often guilty of a passive learning style. This is often blamed on the vast knowledge base of facts that must be mastered; hence students must be told what they need to know. Bloom [1956] speaks of levels of learning. The very basic fact level only accomplishes level 1 in Bloom's taxonomy. Active learning results in higher levels of achievement [Doran 1996].

The curriculum clearly states topical content to cover. Recent work has considered depth of coverage in terms of Bloom's levels [Doran 1995]. Courses successfully introduce the necessary topics of problem solving, teamwork and life cycle issues; but practical experiences are often lacking [Doran 1993], [Daigle 1995, 1996]. Also lacking is a necessary component of human interaction and nurturing. Elliot Soloway stated this recently at ACM'97 held in San Jose. Technology plays a critical role in education today, however, without the human interaction between teacher and student, learning will fail. Soloway states no use of technology can replace this human element. How can the educational experience be personalized? How can educators provide a more individualized approach to learning without sacrificing classroom content? One way to personalize education and to maintain the human element is through *Mentoring*.

Mentoring

Mentoring is hardly a new concept. The word *mentor* comes from Greek mythology. Mentor was the trusted friend of Odysseus and the tutor of Telemachus, Odysseus' son. During the Trojan War, Odysseus entrusted the care of his household to Mentor. The word mentor now refers to a trusted advisor. [Grolier, 1996].

Mentoring within academia occurs both informally and formally. When you take that special student under your wing, looking out for their best interests, guiding them through the curriculum, and looking out for meaningful academic experiences for their development, you are mentoring in an informal way. Other examples of informal mentoring are advising professional organizations such as ACM and AITP, coaching a team for a programming contest, or organizing a faculty-student softball intramural team. There is no formal commitment to involvement but there is a desire to have greater involvement and a deeper understanding of people rather than the roles of student or faculty.

When you analyze a student's plan for next year's courses for relevance to the student's stated academic objectives, you are mentoring in a formal way. Some other traditional ways of mentoring include directed studies participation and thesis preparation. In each of these situations there is a formal commitment to guiding the student to success, either in the curriculum, a course, or to produce a product.

These examples of mentoring illustrate ways that a faculty member can provide a personalized learning experience. However, these experiences are often limited in scope and duration, and, except for advising, are generally isolated incidents that involve few faculty and students. Few long term mentoring relationships are associated with undergraduate education.

Seven years ago a freshman mentoring program was established by the Dean of Students at our university. Volunteers were recruited from among faculty to provide personal guidance to new, academically gifted students from faculty in the student's major during the freshman year. The School of CIS embraced the concept of mentoring and has incorporated in informal and formal activities within its academic community.

The involvement of School of CIS in the university mentoring program has been exclusively through the efforts of one of our faculty. However, the idea of mentoring as a means of becoming involved in the academic development of students has taken root and evidence of mentoring can be also be found formally and informally within the CIS academic community. The CIS faculty member involved with the university-wide program initiates contact with mentes and serves as the primary mentor during the freshmen year. In subsequent years, this faculty member maintains the contact and invites the participation of other faculty and students. Mentes of the university-wide program serve as leaders in these on-going projects. The remainder of this paper discusses the mentoring activities in the freshman year, the informal mentoring activities during the sophomore and junior years, and the formal mentoring during the senior year.

Formal Mentoring in the Freshman Year

New college students face many obstacles to successfully transition from high school to college: There are the external stresses of obligations to family, employment, and financial as well as the traditional stresses of brought on by course work, peer pressure, social interaction, time management, and more independence because of being away from home. Even those who are among the best academically prepared, possessing excellent high school grades and high ACT/SAT scores, are not immune to these obstacles to success. Students who should be academic leaders often fail to perform any better than at-risk students when they succumb to transition stress.

The University of South Alabama created a freshman mentoring program to provide additional support to academically gifted students. For many of these students, this is their first time away from home, the first time they must make decisions on their own. Volunteer faculty mentors are there to nurture these leaders-of-the-future during the embryonic stages of their academic development.

During the freshmen year, students primarily participate in the university-wide program. The Dean of Students has established various programs and activities during that year to integrate these students into the university community. Emphasis is placed on adjusting to college life, study skills and social interactions. There are also educational field trips set-up by various faculty from around the university. Students are strongly encouraged to participate in the activities and network with other students in all disciplines of study.

The CIS faculty mentor participates in these university-wide activities and also establishes local CIS events for the freshmen members of the mentor program. Early in the process, social meetings are used to allow the new mentor participants to meet and talk to previous members. Many of these students take an active role in the local student chapter of ACM or AITP, frequently assuming positions of leadership and involvement in the CIS tutoring program. By providing academic and personal advice, by assisting with decision making, and by involving the student in social and academic activities, the mentor personalizes the educational process for the freshman year and fosters an attitude of involvement and leadership in professional activities.

In the next section we discuss the informal mentoring activities that result in the remaining years in the program from the freshman year relationship. The next section will describe how the faculty within our CIS program has fostered a mentoring environment.

Informal Mentoring in the Following Years

The university encourages continued mentoring but it does not provide funding to support mentoring after the freshman year. Once benefits of freshman year mentoring were realized, the CIS mentor elected to mentor through the remainder of a mente's academic career. Using the freshman experience as a springboard, the CIS freshman mentor and previous mentes extend the notion of mentoring to a larger portion of the CIS community. Three new types of mentoring takes place during this time: the mentoring relationship established in the freshman year is continued, the previous mentes assist in the mentoring of new mentes assigned to the faculty member, and more CIS faculty and students join in the advanced projects and activities.

The CIS mentor and a previous freshman mente initiated a series of web projects that resulted in the involvement of several other faculty and students. The first web project involved the ACM student organization in which both mentor and mentes were active: this project created a demand for similar projects for other student clubs in which mentes were involved. Web projects provide excellent opportunity to explore the challenges of the CIS discipline since they are practical applications for developing a wide range of problem solving skills and extending classroom knowledge. They also furnish an opportunity for hands-on experience with data communications and networks that is useful when seeking employment in today's business sector [Belanger 1995]: network management, client/server architectures, internet protocols, message handling, file transfer, all are involved in these projects.

The CIS mentor assisted with finding computing resources, with recruiting CIS faculty for technical mentors, with managing the ACM project, and with maintaining an overview of succeeding web projects. The mentor emphasized the application of the Software Development Life Cycle (SDLC) for project planning and management and a divide-and-conquer approach to the project implementation strategy. The larger problem was broken down into micro-objectives that could be achieved through a well planned sequence of sub-goals [Gagne 1985, 1988], [Salisbury 1989], and [Doran 1993]. The accomplishment of the sub-goals lead to the completion of the overall goal, establishing overall effectiveness [Doran 1994].

Sophomore and Junior mentes are given significant responsibilities for the project: they interact with the CIS mentor regarding the project goals; they work closely with the CIS faculty technical mentors for project implementation strategies; and they serve as project managers for new recruits. This involvement increases their technical, interpersonal, and leadership skills and it also results in their evolving to the role of mentor. They experience the benefits of giving and the complexities of working with others when they show peers how to apply Polya's problem solving strategy to task assignments and when they share technical knowledge necessary to carry out task assignments

Projects like these offer a practical means of expanding the presence of mentoring within the School of CIS academic community by extending the university freshman mentoring program goals and by increasing participation of additional CIS faculty and students. In the next section, we discuss an in-class approach to mentoring during the senior year.

Formal Mentoring in the Senior Year

The capstone of the CIS undergraduate curriculum is a two quarter sequence that involves team-oriented projects. The objectives, constraints, and student diversity of the sequence result in a challenge for students and a complex management problem for the instructor.

- Projects are intended to bridge the gap between academia and the real world, and, therefore, are of greater complexity and of longer duration than could be achieved in previous classes.
- Accountability for individual activities and for collaboration with team members are major components of individual assessment.
- Individual and group writing assignments and informal and formal presentations are required.
- Students view the sequence as an opportunity to personalize the curriculum: The instructor must balance the short-term student requests with long-term employer expectations.

To achieve course objectives and to satisfy student interests, an independent-team model for course management is used: each team functions as a self-managed team with project problem different from that of any other team. A loosely-coupled association with other members of the class is maintained with two types of events. Weekly class meetings provide an open forum for reporting on team progress and for discussing team strategies; special class meetings provide a means of critiquing formal team presentations before they are made before a public audience.

While the independent-team model provides the flexibility for meeting course objectives and satisfying student interests, it does create another problem. Although a solitary faculty member can be expected to provide guidance for the project management process for all teams, a team's project may require technical or domain knowledge unfamiliar to the instructor. Mentors are recruited, either by a team or by the instructor, to supply the expertise necessary for a successful project experience. During the past two years, twenty-five project teams have formed mentoring relationships with thirty-eight individuals.

Several sources of qualified mentors exist. We are proud of the enthusiastic support from our own CIS faculty: about 85% of our CIS faculty have volunteered as domain and/or technical mentors for project teams. Other university faculty have also been utilized: one project involved faculty from statistics, engineering, and finance. CIS alumnus participation brings the past, present, and future together: four alumni have participated in three separate projects involving new technology. Teams have also received guidance from CIS professionals from a local engineering firm, a printer manufacturer, a real estate agent, a principal of a middle school, and the corps of engineers. These affiliations have provided experiences that otherwise would be unavailable.

Discussion

Mentoring has always had a presence in academia. The university freshman mentoring program and the commitment of one of faculty to mentoring has resulted in an increased presence within our CIS academic community. The benefits of a deliberate approach to mentoring are numerous.

Mentoring establishes an environment for leadership development (for faculty as well as students). We have observed that students who participate in this mentoring process evolve into leaders. These students assume leadership roles in extracurricular CIS activities and on senior project teams. They also are active in student organizations and are a visible presence on campus. The example and leadership provided by these students extends to include many additional students not initially involved with the university-wide mentor program.

A mentor is a role model. Mentoring provides the opportunity to teach character, ethics, and the need for life-long learning by example. Each of us has a story of special help and guidance that we received during our personal and professional development--we succeeded because of nurturing that we received along the way. By their unselfish giving, mentors communicate a responsibility for continuing the tradition of nurturing--for being a good citizen in the profession and in society.

Mentoring results in an individual approach to education and greater involvement of faculty in the academic development of students (the human element). The experiences of being mentored in the freshman year, interacting with other CIS faculty and students on special projects in the succeeding years, and mentoring of other CIS students, result in a broader and more intense development of individual technical, communication, and interpersonal skills. Several of the mentes have joined in with faculty in presentations to conferences.

In an academic environment, a mentor provides insight into learning beyond the traditional classroom setting. We believe that our approach to building mentoring into our academic program, though in-class and out-of-class activities, is an additional way to incorporate the human aspect of learning that Soloway was addressing in San Jose.

References

[ACM 1991] ACM/IEEE-CS Joint Task Force, Computing Curriculum 1991.

[Belanger 1995] F. Belanger and D. H. Jordan, "Assessment of the World Wide Web as a Teaching Tool", Proceeding of ISECON'95, November 1995, Charlotte, NC, pp. 94-100.

[Bloom 1956] B. S. Bloom et al, The Taxonomy of Educational Objectives: The Classification of the Educational Goals, Handbook I: The Cognitive Domain. McKay Press, New York 1956.

[Daigle 1995] R. J. Daigle and H. E. Longenecker, "Successful Application of Principle Centered Leadership in the Information Systems Project Course", Proceedings of ISECON'95, November 1995, Charlotte, NC, pp. 194-200.

[Daigle 1996] R. J. Daigle, M. V. Doran, J. H. Pardue, "Integrating Collaborative Problem Solving Throughout the Curriculum", Proceedings of the 27th SIGCSE Technical Symposium on Computer Science Education, February 1996, Philadelphia, PA, pp. 237-241.

[Doran 1993] M. V. Doran, H. E. Longenecker, and J. H. Pardue, "Strategies for Success in CS1 and CS2: Implications of Polya in Implementing Software Engineering Principles", Proceedings of 11th Annual Conference of the Association of Management, invited talk, August 1993, Atlanta, GA, pp. 1-6.

[Doran 1994] M. V. Doran, H. E. Longenecker, and J. H. Pardue, "A Systems Approach to a Data Structures Course for Information Systems Students Consistent with DPMA IS'90", Proceeding of ISECON'94, October 1994, Louisville, KY, pp. 236-243.

[Doran 1995] M. V. Doran and D. D. Langan, "A Cognitive-Based Approach to Introductory Computer Science Courses: Lessons Learned", Proceedings of the 26th SIGCSE Technical Symposium on Computer Science Education, March 1995, Nashville, TN, pp. 218-222.

[Doran 1996] M. V. Doran, "A Framework to Consider CIS Projects By Non-Majors Entering CIS Graduate Programs", Proceedings of the Americas Conference on Information Systems, August 1996, Phoenix, AZ, pp. 860-862.

[DPMA 1991] Information Systems: The DPMA Model Curriculum for a Four-Year Undergraduate Degree, IS'90, Park Ridge, IL, DPMA.

[Gagne 1985] R. M. Gagne, The Conditions of Learning, Holt, Rinehart and Winston, New York, 1985.

[Gagne 1988] R. M. Gagne, L. J. Briggs and W. W. Wager, Principles of Instructional Design, Holt, Rinehart and Winston, New York, 1988

[Grolier 1996] The 1996 Grolier Multimedia Encyclopedia, Version 8.1S, 1996, Grolier Electronic Publishing, Inc

[Parks 1990] S. D. Parks, "Social Vision and Moral Courage: Mentoring a New Generation", Cross Currents, Fall 1990, pp. 350-367.

[Parks 1995] S. D. Parks, The University As A Mentoring Environment, The Indiana Office for Campus Ministries, Indianapolis, IN, April 1995.

[Salisbury 1989] D. F. Salisbury, "What Should Instructional Designers Know About General System Theory?", Educational Technology, August 1989, pp. 42-45.

An Interdisciplinary Informatics Institute: Organization and Curricula

Z. Onay and N. Yalabýk
Department of Management, Department of Computer Engineering
and
Informatics Institute
Middle East Technical University
Ankara, Turkey

Abstract

This paper presents the rationale and organization of an Informatics Institute to foster interdisciplinary education and research in information systems (IS) and related fields. It proposes curricula for graduate studies stressing the importance of interdisciplinarity in the field and underlines the significance of research and links with industry. It also discusses the role such an institute can take in diffusing information technology (IT) knowledge to wider audiences through new educational technologies, to lay the foundations for a stronger awareness of the field.

1. Introduction

As we approach the 21st century, the rapid development and convergence of information and communication technologies gave rise to a wide and ever growing expanse of information systems applications in almost all sectors of the economy. This prompted leading IS educators and universities into reconsidering the paradigm for equipping their graduates with the skills required to foresee and use the full potential of IS (1). The need is for graduates who are able to grasp the implications and impact of IS in the context of their activities, and recognize that information is a valuable resource and exploit it as such. Technical specialization is no longer sufficient and needs to be reinforced with a broader spectrum of knowledge to deliver effective solutions to real world problems. The fast changing, multidisciplinary nature of the subject presents a real challenge in determining the best way to respond to this need.

Informatics is a field where the theory and practice of a number of disciplines merge. It embraces and includes the area of IS. Considering the array of application areas, ranging from agriculture, government, manufacturing, finance, healthcare to arts, *informatics* should inherently be accepted as an interdisciplinary field. It represents the amalgamation of computer science, communications, management and a complementing area of application (Figure 1). An example is Medical Informatics, which is an accepted field in itself and covers areas such as Information Systems, Image and Signal Processing, Decision Support Systems as well as Medicine (2).

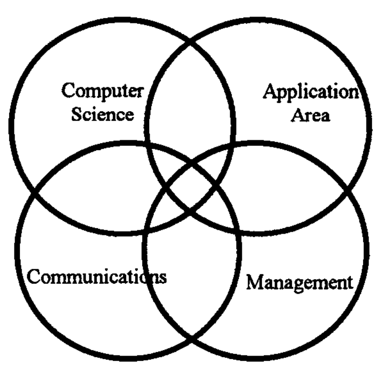


Figure 1. Informatics and Interdisciplinarity

Interdisciplinary informatics programs have already been developed in many universities worldwide. There is a wide variety in the approach in terms of courses taught, degrees offered and implementation options. Examples are the School of Informatics and Sciences at Nagoya University (Japan), The Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign, the Information Networking Institute at Carnegie Mellon University, the Systems Design and Management Program at MIT. All of them offer graduate degrees and thus combine education with interdisciplinary research. As in these examples, the challenge at the graduate level can be approached by forming a unit that integrates many possible interdisciplinary informatics areas such as IS, Cognitive Science, Medical Informatics, Robotics, Scientific Visualization, under the same roof but independent of faculties and departments. This requires the breaking down of barriers between disciplines since there are many overlaps. Careful consideration has to be given as to which organization structure to adopt to facilitate interdisciplinary education. Clearly the support and affiliation of many departments, such as Computer Science, Electrical and Electronics Engineering, Management, are needed. Several attachments exist and the choice is of prime importance for long term sustainability. The main factors affecting the final choice of a single, independent umbrella unit can be cited as:

- the ability to obtain the support of the wide-ranging related faculties and departments in delivering the courses in the programs;
- the coordination necessary to keep and maintain the curricula up-to-date;
- the need to foster interdisciplinary research;
- the fair allocation of resources to supporting departments and faculties;
- the ability to extend the scope of services to training courses offered to faculty and administrative staff within the university and to outside groups like executives;
- the need to have an independent identity for the efficient use of university resources and more importantly the necessity to uphold the ownership, coordination and balance between programs.

The remainder of the paper will discuss the scope, programs and activities of the 'Informatics Institute', which has the affiliation of several departments within the University.

2. The Informatics Institute

The Institute has three major goals: the development and delivery of interdisciplinary graduate programs in informatics, the activation, coordination and support of pure and applied research in the field, the establishment and maintenance of strong links with industry. In addition, the Institute has the mission of propagating IT knowledge to all disciplines within the University.

2.1 Interdisciplinary Graduate Programs

The Institute's programs are designed to ensure interdisciplinarity without sacrificing the depth required at the graduate level to the breadth of knowledge necessary in informatics. The objective is to enhance communication and give graduates a better appreciation of other disciplines, a broader vision to be more effective in the workplace. Initially, three programs, leading to the degree of Master of Science are being considered: the Information Systems Graduate Program, the Cognitive Science Graduate Program and the Human-Computer Interaction Graduate Program. The curriculum development activity for the first two has been completed. The curricula are presented in Appendix 1 and Appendix 2 respectively.

2.1.1 Information Systems Graduate Program

The program will expose students to current issues in IS and give them a practical understanding of the technology-oriented business opportunities. The curriculum draws faculty from the departments of Computer Engineering, Management, Industrial Engineering and Electrical Engineering. The aim is to prepare students to be proactive in making the most of the competitive leverage offered by information systems in their working environments.

2.1.2 Cognitive Science Graduate Program

The program covers topics in Artificial Intelligence, Computer Science, Linguistics, Psychology, Cognitive Neuroscience and Philosophy. The aim is to emphasize the cognitive aspects of computational approaches to speech and language analysis, teaching, and innovative and creative uses of computers in an information based society.

2.1.3 Human-Computer Interaction Graduate Program

This program has a very wide spectrum, covering areas from scientific visualization and virtual reality to all educational tools using computers such as asynchronous learning. Students in this program learn to create these tools, depending on the area of application.

2.2 Widespread Informatics Education: Asynchronous Learning

An objective of the Institute is spreading IT knowledge to wide audiences. Plans are required on how to accomplish this at all levels; university wide and maybe nationwide. The realization of this objective with classical teaching methods is very difficult, considering the size of the target student population and the limitations on the availability of teaching staff. On the other hand, teaching IT necessarily involves working with a computer most of the time. Thus, the use of new educational technologies, namely, asynchronous learning through Internet is proposed as an efficient and effective way in attaining such a goal.

Asynchronous learning is in the process of being accepted worldwide and examples are many, like the SCALE Programs of the University of Illinois at Urbana-Champagne, the Information Systems Graduate program available through CIGNA at Drexel University, the GEMBA Program of Duke University, and the program of NASA. Asynchronous learning for programs and courses are being implemented in many universities campuswide, nationwide and even worldwide (3).

Although Global Departments and even Global Universities are spreading extremely fast and many examples of IT teaching are available on Internet, the feasibility of a unified approach that will cover the whole field and the organizational aspects of doing this on a very large scale needs careful consideration. Institutes of the type proposed in this paper, may also take the initiative in doing this, especially as widespread IT education is also aimed at teaching IT beyond simple computer literacy and thus, serves as a preparation for interdisciplinary studies.

2.3 Research Topics

Today, interdisciplinary research plays an important role in economic development since many revolutionary ideas are developed by those with a breadth of vision as well as depth, which can only be attained by working with people with different perspectives and knowledge. Many universities are restructuring their programs to make them more interdisciplinary (4). Interdisciplinary research in informatics is a major activity of the proposed Institute. Interdisciplinary research topics can cover an extremely wide spectrum, from image processing to strategic information systems or microelectronics (5). The Institute provides support to projects that bring people together from corresponding disciplines. The focus is on specific areas to ensure the efficient utilization of manpower and infrastructure.

The Institute concentrates its efforts in selecting research areas and finding support for these. Areas might change according to the interest profiles of the available academic staff but most value is obtained by initiating interrelations among people from different parts of the campus. This will hopefully lead to the birth of interesting and fruitful research efforts.

2.4 Industry Relations

It is an accepted fact that no academic unit can work isolated from the real world, especially in a dynamic field like informatics. Continuing relations with industry are very important in designing curricula, choosing research topics, finding lecturers and obtaining support from industry. Lasting links can be formed only if the counterparts from industry feel that they are part of the whole process of design and implementation of the programs. A good way of achieving this is to form permanent committees that bring together academicians and industry leaders and get active feedback at each stage of the process. The members of the committee should be able to grasp the technical aspects as well as the restrictions of both sides. The Institute embodies such a committee which meets regularly to explore different ways of cooperation. The committee also acts as an advisory board for the Institute to remain responsive to the needs of industry.

3. Conclusion

The paper presented the scope, organization and activities of an Institute for research, development and education in interdisciplinary informatics. The importance of interdisciplinary studies has been stressed. Doubtless, the approach carries certain risks. Ensuring the continuing support of affiliated departments, mediating the territorial instincts of each discipline, avoiding inertia and stagnation are significant issues. However it can be claimed that the chances of success for such undertakings are great if people can learn to listen to those who have a completely different perspective.

References

1. Cohen, Eli Boyd. "Evolving Trends in Information Systems". Journal of Information Systems Education. 5(4), 1993.

2. Yearbook of Medical Informatics 95. Publications of IMIA, Schattauer, 1995.

3. Hiltz, Starr Roxanne. "Impacts of college-level courses via Asynchronous Learning Networks: Focus on Students". Sloan Conference on Asynchronous Learning Networks, Philadelphia, 1995.

4. Director, Stephen W., Khosla, Pradeep K., Rohrer, Ronald A., Rutenbar, Rob A. "Reengineering the Curriculum: Design and Analysis of a New Undergraduate Electrical and Computer Engineering Degree at Carnegie Mellon University". Proceedings of the IEEE. 83(9), 1995.

5. Bell, Trudy E. "The Beckman Institute for Advanced Science and Technology". IEEE Spectrum. November 1996.

Appendix 1

Information Systems Graduate Program Curriculum

(Thesis and Non-Thesis Options)

Deficiency Courses

Introduction to Computers and Programming
Algorithms and Data Structures
Introduction to Probability and Statistics
Fundamentals of Business
Introduction to Data Base Management Systems ***(thesis option)***
Software Engineering ***(thesis option)***

Core Courses

Information Systems Management
Information Systems Project
Data Communications and Computer Networking
Graduate Seminar
Introduction to Data Base Management Systems ***(non-thesis option)***
Software Engineering ***(non-thesis option)***
Term Project ***(non-thesis option)***
Master Thesis ***(thesis option)***

Area Tracks (one course at least from each track)

Software Systems

Software Quality Assurance and Testing
Database Management Systems
Design and Analysis of Algorithms
Multimedia
Design, Development and Evaluation of Instructional Software
Fundamentals of Image Processing
Object-Oriented Analysis and Design of Information Systems
Object-Oriented Programming Languages and Systems
Artificial Intelligence
Neurocomputers
Knowledge-Based Systems

Management Systems

Network Flows and Project Management
Management Science
Organization and Management
Management Information Systems
Human Factors in Information Systems
Decision Support Systems: Design and Implementation
Regulatory and Legal Aspects of Information Systems
Business Process Redesign
Total Quality Management
Applied Optimization
Applied Stochastic Modeling
Managerial Economics
Human Resources Management
Creativity
Negotiation Process
Topics in MIS
Systems Simulation

Communication Systems

Distributed Computing Systems
Information Theory
Computer Security and Cryptography
Guided Wave Optics
The Politics of Telecommunication, Regulation and Standardization
Computer Networks and Communications
High Speed Networks
Coding Theory
Communication Theory

Degree Requirements

Thesis option: 7 courses, Graduate Seminar, M.S. Thesis
Non-Thesis option: 10 courses, Graduate Seminar, Term Project

Appendix 2

Cognitive Science Graduate Program Curriculum

Deficiency Courses

Introduction to Programming
Lisp and Prolog
Formal Languages
Logic
Introduction to Linguistics

Area Tracks (one core course from the compulsory courses list of each track)

Computer Science

Compulsory Courses
Artificial Intelligence
Computational Linguistics I
Neurocomputing

Elective Courses
Logic for Computer Science
Cognitive Aspects of Natural Language Processing
Computers and Commonsense Reasoning
Machine Learning
Varieties of Formal Semantics
Advanced Neural Modeling
Computational Vision
Pattern Recognition
Image Processing

Linguistics

Compulsory Courses
Modern Theories of Grammar
Cognitive Linguistics
Psycholinguistics
Applied Linguistics

Elective Courses
Theoretical Linguistics
Language Acquisition
English-Turkish Contrastive Analysis
Pragmatics and Discourse

Psychology

Compulsory Courses
Cognitive Processes I
Cognitive Processes II
Visual Perception

Elective Courses
Introduction to Cognitive Science
Psychoacoustics

Philosophy

Compulsory Courses
Topics in Epistemology
Philosophical Logic I
Philosophy of Mind I
Theories of Scientific Method

Elective Courses
Philosophical Logic II
Philosophy of Mind II
Philosophical Foundations of AI
Philosophy of Language
Dynamics of Scientific Theories
Philosophy of Science

Degree Requirements: 8 courses, Graduate Seminar, M.S. Thesis

Adopting Non-Budgetary Methods For The Construction Of A Client-Server Teaching Laboratory

Mayur R. Mehta, Southwest Texas State University, San Marcos, TX 78666-4616
George W. Morgan, Southwest Texas State University, San Marcos, TX 78666-4616
Stanley D. Stephenson, Southwest Texas State University, San Marcos, TX 78666-4616

Abstract

Academic departments in the field of computer information systems (CIS) are continually developing strategies to pull the faculty and student into higher levels of quality education. The impetus for such a pull has been the understanding that, in order to remain competitive and to prepare the CIS student for their first job, changes in the curriculum and provisions for enhancing current computing facilities are both necessary and paramount. Since the CIS student is learning an applied discipline, the technology induced push, along with a close advisory relationship with the CIS industry, will often cause this process to spiral out of budgetary control. Revising training programs and acquiring new technology may be accomplished by adopting other non-budgetary methods. This paper will explore one academic department's process of constructing a client-server teaching laboratory. The process involves first assessing the department's future computing facility needs and then implementing a strategy for total environmental involvement by forging a partnership with industry, vendors, alumni, and departmental faculty.

Introduction

The perceived gap between the technical skills and tools the CIS curriculum offers and what industry expects the graduates to have upon graduation has been well documented. According to Hensel (5), this gap forces the CIS departments to address four major issues:

1. The curricula (DPMA, ACM, etc.) that should be used to guide the programs offered by universities for CIS Majors.
2. The technology training that CIS departments should offer.
3. Help and support the industry should provide to ensure a supply of trained, skilled CIS graduates.
4. The means for acquiring the latest technology for the CIS degree programs when budgets are shrinking.

Several earlier works (1,2,3,4,6,7; 8) have provided an in-depth analysis of the first three issues. This paper's primary focus is to provide one CIS department's approach to resolving the issue of identifying and acquiring the technology to support a modern, state-of-the-art CIS curriculum. The department would first attempt to identify the current level of technology being used by corporations that recruit its graduates. It would next determine whether or not the identified level of technology could be classified as a fad. Finally, the cost and means by which this technology will be acquired would be identified. Several factors were considered in identifying the level of technology in use.

The first consideration was to ensure that graduates were trained on and exposed to technology that they were most likely to require as they moved from an academic environment into their entry-level positions. In other words, the market should drive the technology selection decision. The second consideration was to ensure that the department focused on technology that was mature and one that would be around for a while. This was to ensure that the computing facilities and curriculum did not face quick obsolescence. This would establish an infrastructure to help make curriculum changes that were proactive and fully supported by the available technology. In short, the graduates would have the training, skills, and exposure to technology required by recruiting corporations.

We took just such an approach as we moved our CIS curriculum from a traditional mainframe-based computing environment to a client-server and networked environment. Our goal was to have new curriculum supported by a computerized teaching laboratory. This lab would be used to enhance instruction in courses such as client-server applications development, networking, advanced system design, decision support systems, and Object-oriented technology.

In the Fall of 1991, the department's advisory council indicated that a microcomputer network or client-server laboratory should become a high level priority need for the department. The council which is composed of middle management representatives from private and public organizations within the State recognized that both client-server computing and networks were rapidly becoming major areas within the Information Technology (IT) field. The council also recognized that graduate of a CIS program should be exposed to the newer emerging environments.

The Assessment Process

We began by identifying the current and potential recruiters of our graduates. Some of these companies are represented on our CIS advisory Council. Using a questionnaire we surveyed senior management and IS development staff to ascertain the current level of technology in use in their organizations. Since the curriculum focus was to be on client-server computing and networking, the survey took a three-pronged approach to identifying the necessary components of a good teaching lab. The survey was designed to identify the client and server hardware platforms, network operating systems and communication media, as well as client-server applications development tools (including database management systems).

While there was a good mix of mainframe, midrange and personal computers in use, the majority of the organizations used mainframes and midrange computers as database servers and for complex computing. Some of the midrange and most of the high-end CISC- and RISC-based computers were used as intranet servers as evidenced by the overwhelming use of Novell Netware and Windows NT as network operating systems.

The client platform of choice was Intel-based personal computers. UNIX-based machines were a close second. Most of the UNIX boxes were either used as servers or as clients in engineering and design related applications. DOS/Windows 3.1, Windows 95 and Windows NT Workstation dominated as the client operating systems of choice among the Intel-based clients.

10BaseT cable was the most frequently used communication media. Moreover, Ethernet LANs were more common than token-ring. Additionally, most of the companies surveyed had internetwork setups using ISDN and FDDI.

Finally, a survey of client-server applications development tools revealed that only about 26 out of the 86 tools available in the market were being used by our recruiters. Out of these 26, PowerBuilder and Visual Basic were cited as the client-server tools used most often, with Visual C++ and Lotus-Notes not too far behind. If you combined all the Smalltalk-based tools in use (Smalltalk, VisualAge, VisualWorks, and VisualGen) then the use of Smalltalk in developing client-server applications surpassed the use of Visual C++ and C++.

With regard to database management systems (the third part of the C/S equation), our survey indicated that SQL-based database servers dominated the scene. These included MS-SQLServer, Sybase SQLServer, and Watcom. Of these, MS-SQLServer and Sybase were the most commonly used. Oracle and Rdb came a close second. On the desktop DBMS side, MS-Access was by far the most often used product. The use of DB2 and DB2/2 was also high as indicated by the large number of mainframe and midrange systems in use.

Implementation Strategy

In the Spring of 1992, the department put together a proposal for a microcomputer teaching laboratory. The department was able to identify the current level of technology in use by current and potential recruiters based on the information obtained from our assessment analysis. This enabled us to develop a proposal for the hardware and software needed to setup the teaching lab. Based on this analysis, the department identified the minimum requirements and constructed the specifications for a client-server teaching lab as shown in Figure 1.

Figure 1

Client-Server Teaching Lab Specifications

1. <u>Server</u>: One Pentium-based with 64 Mb and 1 Gb hard drive
2. <u>Clients</u>: Thirteen 80486 or better with 16 Mb and 15" monitors (including Instructor's console)
3. <u>Media</u>: 10BaseT
4. <u>Topology</u>: Bus (physical configuration Star)
5. <u>NOS</u>: Novell Netware 3.12
6. <u>Client OS</u>: DOS/Windows 3.1
7. <u>C/S Tools</u>: PowerBuilder, VisualBasic, Smalltalk
8. <u>DBMS</u>: SQLServer, Oracle, and Access
9. <u>Printer:</u> Laser Printer with at least 10 PPM.

The proposed lab design, which had an estimated cost of $50,000 (1992 dollars), was then presented to the council in the Fall of 1992. The council was in full agreement with the proposed lab design and basically wished the department well in getting the lab funded. As it turned out, there was absolutely no financial support available from the companies represented on our advisory council or the University. The lab proposal lay dormant for the rest of the 1992-1993 academic year until available from the companies represented on our advisory council or the University. The lab proposal lay dormant for the rest of the 1992-1993 academic year until the project was identified as a high level goal of the new acting chair of the department in the Fall of 1993. Rather than relying on the traditional means of seeking funds from the University and the State, a three-fold plan of attack was immediately implemented in order to obtain donations from a variety of non-budgetary sources.

First, a commitment from the Dean of the School of Business was sought in order to obtain a room to house the lab; this was contingent, however, upon the acquisition of all necessary lab equipment. Because the lab needed physical space, this request was made early in the Fall 1993 semester. If a room could not be made available, there would be no need to move forward with the proposed lab.

Second, an aggressive donated funds program was started. This program was aimed at three groups: (1) members of the CIS Advisory Council and the corporations that they represented, (2) CIS alumni and faculty, and (3) other sponsors. At the Fall 1993 meeting of the CIS Advisory Council, it was acknowledged that while no one organization could fully fund the proposed lab, several organizations, each donating a smaller amount, could make it a reality. It was requested that the council members initiate inquiries within their organizations to solicit for money to get the lab started. The chair also initiated requests to other organizations who were not members of the Advisory Council but who were supporters of the CIS program or who were involved in the IT field. All CIS alumni were contacted by letter and invited to help with the lab. As a way of incentives to the alumni, the department offered T-shirts for one level of support or names on a plaque to be permanently displayed in the lab for a higher level. In addition, the CIS faculty was asked to help with the lab. One faculty member donated a 386 PC to the facility while half of the faculty donated monetarily.

Third, an equipment acquisition plan was prepared and implemented. Once again, the CIS Advisory Council and other sponsors were contacted in order to assess their ability to provide any equipment that they had available which would match the overall design and plan of the proposed network lab. These three actions initiated during the Fall 1993 semester produced several good results; without which, the project would have been left on very tenuous ground.

Lab Construction

Fortunately, the Fall of 1993 was an ideal time to request space. Enrollment in the School of Business had been declining for the past six years, along with a corresponding decline in faculty. The School had a room that had housed several instructor-level faculty, most of whom had since moved to regular offices. This room had been slated to be returned to classroom status but converting it to a computer lab was actually more advantageous to the School. Consequently, the dean promised that if the department could obtain the equipment and software for a computer lab, he would allocate that room to the department. This commitment was obtained early in the Fall 1993 semester. Moreover, a small amount of money had already been set aside for renovation; this money was subsequently used to convert the room into a computer lab.

The requests for funds produced mixed results. Several CIS Advisory Council members were able to obtain monetary donations from their organizations; these moneys started arriving in late 1993 and early 1994. Contributions ranged from \$1,000 to \$4,000. Only a small number of alumni donated money, but those who did were very generous. In particular, we benefited from our alumni who were working for companies that provided matching donations. In one case, six alumni who worked for a major oil company donated \$600 that was matched three folds by the company. In other words, these six alumni produced a total of \$2400.

There were two by-products of our alumni campaign. One alumnus, who had a small networking company and was Novell certified, indicated that for his donation he would be glad to do all the wiring and initial system setup for the client-server lab. In addition, he offered to provide several components and some ancillary software. His contribution was subsequently valued at over \$2000. Another alumnus worked for a PC manufacturer and offered to sell us PCs at cost. These two sources of help were unexpected but much appreciated.

The request for used network and microcomputer equipment was one of the most positive aspects of the lab development. Within one month of the Fall 1993 CIS Advisory Council meeting, one insurance company donated six 386 PCs and an oil company donated the networking hardware. The fact that these donations occurred so quickly was a tremendous boost to the department. It demonstrated that assistance for a lab was available. The only major problem was still identifying where that assistance might be found.

By December 1993, the department was convinced that the lab could be developed by the start of the Fall 1994 semester. However, three major problems remained. First, the cost of the file server required to run the network was prohibitive. It was estimated that such a file server would cost about $8,000 in 1993. Second, the proposed lab needed to have at least 12 workstations, and only 7 were available (the 6 donated by the insurance firm plus the one donated by a faculty member). Five more workstations would cost somewhere between $7,500 and $10,000. Third, decent networking software was going to cost about $2,500. Moreover, site licenses for applications software would cost in the neighborhood of $8,000 particularly since many of the software vendors (e.g., Oracle and Powersoft) did not have academic programs. There was not enough money to solve all three problems at the same time.

Early in the Spring 1994 semester, the first major purchase was made by the department: an instructor's console. Although this console was not needed at that time and in fact would have been less expensive later, its presence along with the seven donated PCs was physical evidence that the lab was under way. Moreover, the instructor's console would have served as our file server in case we were unable to acquire the server specified in the proposal through other sources. Because the room for the lab was not yet ready, the console and the seven, on-hand workstations were placed (but not networked) in the back of one of the classrooms where CIS majors and faculty could see them. Again, the idea was to establish an early physical presence and develop the perception that a client-server lab was under construction.

The single most important act in terms of the lab actually being developed occurred later that Spring. After several months of negotiations, the chair was able to obtain a commitment from a major PC manufacturer to donate a server and four workstations. Moreover, these were high quality items. For instance, the server was a Pentium 100 with a 1 Gb SCSI hard drive and a CD-ROM. This PC was almost state-of-the-art at that time. The workstations were all 486s with 16 MB of RAM and 15 inch monitors. Without this donation, the lab might not have survived the setup stage.

These items were to be donated by August 1994. Once the server and four workstations were added to the already donated equipment, there was enough money to purchase a printer, Novell Netware 3.12 network operating system, and other software. By August, the lab was beginning to take shape; it would have the minimum required 12 workstations networked to a reliable file server.

The decision was then made to schedule two classes for the Spring 1995 semester. The rationale was that we would need the Fall 1994 semester to install all of the application software for the scheduled classes as well as ensure that the setup was properly handled. As the lab became more of a reality, previously unknown costs (e.g., furniture, power strips, wiring connectors, etc.) were popping up.

A one-course reduction for the Fall 1994 semester for the faculty member tasked to teach the major new lab course (client-server applications) was obtained. This reduction permitted the faculty member to set up the lab for the scheduled classes and prepare for a course that had never been taught before on the campus.

Once the University realized that the department was going to have a client-server lab, it provided sufficient funds to purchase tables and chairs. Although the workstations we had at that time were of questionable quality, we knew that eventually we would have a better-equipped lab. Consequently, the decision was made to purchase high quality computer tables and ergonomic chairs. We wanted the CIS majors to see the lab as something special; we also had plans to offer continuing education courses in the lab to generate funds for upgrades.

A Working Client-Server Lab

In August, the department took delivery of the server and four workstations. The CIS graduate previously mentioned returned to campus and spent two additional days installing Netware 3.12, which we had purchased under an academic license. On September 1, 1994, we had an operational client-server lab. Although it was not of the quality we had desired (it was a combination of 386 machines with 2-4 MB RAM and 486 machines with 16 MB RAM), it did work. In fact, it was probably more typical of networks than we realized. In addition, it looked good.

The first courses were offered in the Spring 1995 semester. That semester was as much a learning experience for the faculty as for the students. For most of the students, it was the first time they had encountered a networked environment using GUI-based clients. For the instructors, it was the first time they had taught on a network. The varying workstation performance levels were a problem but not one that could not be overcome. From day one, the students took to the client-server lab. That semester every recruiter who came to campus talked about how the students they interviewed were very proud of the lab.

client-server applications; without the lab, we simply could not teach that course. The lab has also permitted our advanced systems course to use such software as PowerBuilder and Visual Basic.

Equally important, the lab has become a place where junior and senior CIS majors go to work on their group projects and on their programming assignments. We also have one VAX terminal in the lab and offer COBOL tutoring to our sophomore students. We want them to see the lab early in their careers. Now that we use Visual Basic in our very first programming class, we also offer Visual Basic tutoring in the lab. We view the lab very much as a recruiting tool.

The quality of the lab grew much more rapidly than we anticipated. The department took the necessary steps to add course fees to relevant CIS courses. Money continues to be donated and support drifts in from unexpected sources. For example, one member of the CIS Advisory Council, who works for a state agency, could not offer any financial support. However, he was the treasurer for a computer related club, which decided to disband. It had $2,200 in its account. When the officers asked what they wanted to do with that money, our Advisory Council member was quick to suggest our lab. We bought two new workstations with that money.

Three years later (Fall 1996), we have a server now with two 1-Gb hard drives and 64 MB of RAM; an instructor's console; and 14, 486 workstations all with 16 MB of RAM. We have purchased more software. The currently enrolled students do not remember a time when we did not have a lab. A lot has transpired in three years.

Summary

This paper has explored one academic department's process of merging its CIS program into a technological infrastructure designed to enhance its overall educational effectiveness. The reason for such a strategic pull was to insure that the emerging skills and tools taught to the CIS undergraduate mirrored what the IT industry was in fact using. To accomplish this end, the department was forced to adopt non-budgetary methods to construct a client-server teaching laboratory when traditional methods of academic funding proved to be inadequate. The process was not so overwhelming that other departments faced with similar fiscal constraints could not rise above the technology challenges of the future.

References

1. Currid, Cheryl. "Is Curriculum Continues To Evolve At The Corporate Level." Infoworld, June 1992, 64.
2. Goldman, James E. "Design and Implementation of a Practical Business Oriented Undergraduate Data Communications Curriculum." Journal of Information Systems Education, Summer 1994, 80-85.
3. Harris, Albert L. "Are We Preparing Our Students for the Real Job Market." Proceedings: ISECON '95, November 1995, 145-150.
4. Heiat, A., N. Heiat, and J. Spicer. "A Empirical Study of Practitioners' and Academics' Views of MIS Curricula." Interfaces, 15(3), 1993, 27-31.
5. Hensel, Mark. "Closing the Gap Between Academic Degree Programs and Industry Expectations." Proceedings: ISECON '95, November 1995, 70-76.
6. Katz, A. I., and R. L. DeMichiell. "Information Systems Undergraduate Curricula: Trends for 1990s." Interfaces, 14(1), 1992, 26-31.
7. Pick, James B., and K.D. Schenk. "A framework for IS Programs and business relationships." Proceedings: ISECON '95, November 1995, 62-69.
8. Welch, Douglas E. "Colleges and Universities Must Alter Strategies for the Changed Job Market." Infoworld, October 1992, 51.

Hands-on LAN Lab for Training Network Administrators

Young J. Kim, Ph.D.
Department Computer Technology
Purdue School of Engineering and Technology, IUPUI
723 W. Michigan St., Indianapolis, IN 46202
(317)274-6797 kim@engr.iupui.edu

ABSTRACT

This paper discusses the experiences and findings from a capstone course, *Design and Implementation of Local Area Networks,* in a sequence of computer networking courses. The first course was offered in Spring 1996 based on NetWare 3.12; a second, in Spring 1997 based on Windows NT Server 4.0. Minimum lab computing facilities for such a course are discussed. Students were divided into teams, and each team had a leader with network administration experience. The paper discusses the objectives of the course, sketches NT Server 4.0 services and features, and outlines the steps of setting up a Local Area Network, while clarifying NT concepts such as domain and trust relationships. Screen images taken from a primary domain controller are used in the paper. On completion of this course, students would be prepared to setup, manage, and support multi-operating system enterprise networks using on-line, context sensitive help facilities, and to understand imminent technologies such as the next generation Internet protocol standard, IP version 6.

1. Introduction

In 1995, the Computer Technology Department of Purdue School of Engineering and Technology, Indiana University - Purdue University at Indianapolis, initiated efforts to develop an instructional Local Area Network (LAN) lab. The details of the efforts were reported in a paper, Renewal of Network Systems Curriculum with Emphasis on Students' Lab Projects.[1] A hands-on LAN setup course, CPT402: *Design and Implementation of Local Area Networks*, was offered in the Spring 1996 semester; the course focused on installing a Network Operating System (NOS) on a server, supporting WorkStations (WS), installing application packages, and creating and managing user accounts. This course was the culmination of a sequence of Computer Networking courses. The course topics included *network configuration, topologies and protocols, network performance analysis, installation of networks, monitoring of networks*, and *security*; these topics are listed for *IS '95.6 - Telecommunications* course of Information Systems in IS'95 Model Curriculum and Guidelines.[2] In an internetwork computing environment with multiple servers, several hundreds of client workstations, and multi-protocol stacks, the functions of system administration and network management are very complex. Different classroom facilities other than those used for teaching programming, database design, or software engineering are needed to train systems administrators. Students need a lab with dedicated computers and networking devices without any restrictions on security.

Performance of micro processors has been doubled "every 15 months" since 1970 while costs have stayed consistant;[3] PCs have been upgraded periodically for supporting new software with improved features. As a result, old used PCs were plentiful on the campus; they were utilized for hands-on computer networking experiments that did not require high performance CPUs.

2. Lessons from the First Hands-on Local Area Networks Lab with NetWare 3.12

For the development of a LAN lab, we had to select a NOS among several candidates. In 1995, NetWare was the most popular NOS on the campus; NetWare 3.12 was stable and available for use in the lab.

During the summer of 1995, used PCs, original PC documents, materials on the Network Interface Cards (NIC) on the machines, software including the NetWare 3.12 package, MS-DOS version 6.21, and MS Windows 3.11 package including all manuals were acquired. Among numerous, NetWare reference books, *NetWare Professional Reference*[11] was selected as the textbook for the lab course. Three new NICs were purchased together with about twenty Bayonet-Neill-Concelnan cable terminators and some T-connectors. Conventions for the server *name, internal network number,* and *network address* for the NetWare Local Area Networks of the lab were established in advance. Prior to the start of the Spring 1996 semester, a prototype LAN with a single server and four workstations was implemented for a feasibility study of the hands-on LAN lab development. The goal of the study was to check the compatibility of IBM PS/Value Point, Personal System/2 Model 55 SX, old NICs, and NetWare 3.12. The lessons from the prototype setup are summarized in the following list:

(1) All parts and information required for performing each step of Network Operating System installation must be ready and documented prior to initiating the step.
(2) It took about one hour and thirty minutes to setup a NetWare 3.12 server on IBM PS/Value Point model 466DX2/D.
(3) Eight MB RAM was not enough for the server - sixteen MB RAM was required.
(4) It took about fifteen minutes to setup a DOS client workstation.
(5) To install the NOS package, every step must be taken precisely as specified in the document.

In the Spring 1996 semester, twenty-nine students registered for the hands-on LAN lab course. Several students had system administration experiences with NetWare LANs; they were interested in learning the principles behind LAN administration. All of the students were familiar with the Open Systems Interconnection reference model and concepts on network protocols such as TCP/IP; some students, however, did not have any hands-on experiences with PC networking. The students were divided into five teams, because only five nodes of IBM PS/Value Point were available for servers. Each team selected a leader with NetWare experience. Every team completed following tasks: checking the status of old PCs, upgrading NIC drivers if required, physical LAN setup with selected components, NetWare 3.12 installation, installation of application packages such as WordPerfect, creation of user accounts and their default directories, and setting up client nodes. Throughout the semester, network trouble shooting was annoying to the students as well as to the instructor.

Experiences and findings from the first hands-on LAN lab course are summarized in the following paragraphs:

(1) Connectors, terminators, and cables were the source of problems. For a LAN with three nodes, four connectors on two sections of the cable, three T-connectors, and two terminators were required - not counting the connectors on the PCs. It was very time consuming to isolate a defective component: an ohms/volt DC tester showed no faults on any components; however, when a terminator or connector was replaced with a new one, the problem was resolved.
(2) New NICs did not cause any problems. Some problems with old NICs were eliminated when their drivers were updated.
(3) Old PCs were adequate for a hands-on LAN lab. For NOS installation, applications package installation, and user account and file security administration, old used PCs with 16 MB RAM and a disk drive with about 300 MB storage space were adequate for servers in the student lab.
(4) Three to four students were adequate per team. One performed the required operations, one or two documented the operations and system responses, and another checked the accuracy of the operations and their sequence against the overall documented procedure.

Every team developed a project plan with multiple phases and progress reports. The instructor reviewed the documents for possible improvements and accuracy as they were progressed. Every team completed a final project report; every student completed a form on the performance of team members for the project.

3. Objectives of the Second Hands-on LAN Lab

Computer networking is one of the explosively expanding and changing technologies. To help the students be prepared for the computer networking job market, the following five objectives were established for the hands-on LAN lab. For the students:

(1) to become familiar with the principles of computer networking behind the *easy-to-use* NOS installation and support procedures,
(2) to master the skills for using current Graphical User Interface (GUI) for setting up a LAN, configuring networking devices and protocols, and monitoring the performance of the LAN,
(3) to become familiar with the customer support mechanisms deployed by the vendors of network components,
(4) to feel comfortable with the explosively evolving technology of computer networking, and
(5) to be ready to embrace new emerging technologies and standards.

For the students to achieve first two objectives as much as possible, the NOS of the hands-on LAN lab should support a broad spectrum of current networking technologies. Numerous reviewers of industry products reported that Windows NT 4.0 was a good candidate for such a NOS. Whenever NIC problems were encountered, students were encouraged to check the Web pages of the products and to download new drivers from the database of the vendors. For new products and emerging new standards, the Web pages are the best resources; in particular, all of the developments on the next generation IP, IPv6[10], are reported in the Web pages. Students were suggested to read Request For Comments (RFC) listed in the IPng Specification.[13]

4. Hands-on LAN lab based on Windows NT 4

Windows NT is a "processor independent, scaleable, preemptive multitasking, and multithreading network operating system" with refined services: Dynamic Host Configuration Protocol (DHCP) for dynamically assigning TCP/IP configuration to clients, Windows Internet Naming Services (WINS) for registering and resolving computer names of NetBios clients on TCP/IP network, and Domain Name System (DNS) service for mapping host names to IP addresses.[4-8]

During the summer of 1996, Microsoft mailed Windows NT Server and Workstation version 4.0 beta 2 CDs to the department. (For the previous NetWare LAN lab, we did not need any CD-ROM drives.) As the first step of Windows NT LAN lab development for the Spring 1997 semester, we decided to purchase CD-ROM drives for the server nodes in the lab. A student who was familiar with Windows NT administration installed a CD-ROM drive on an IBM 486DX2/66, and setup the beta 2 NT server on the computer.[9] The textbook selected was *Mastering Windows NT Server 4.*[4]

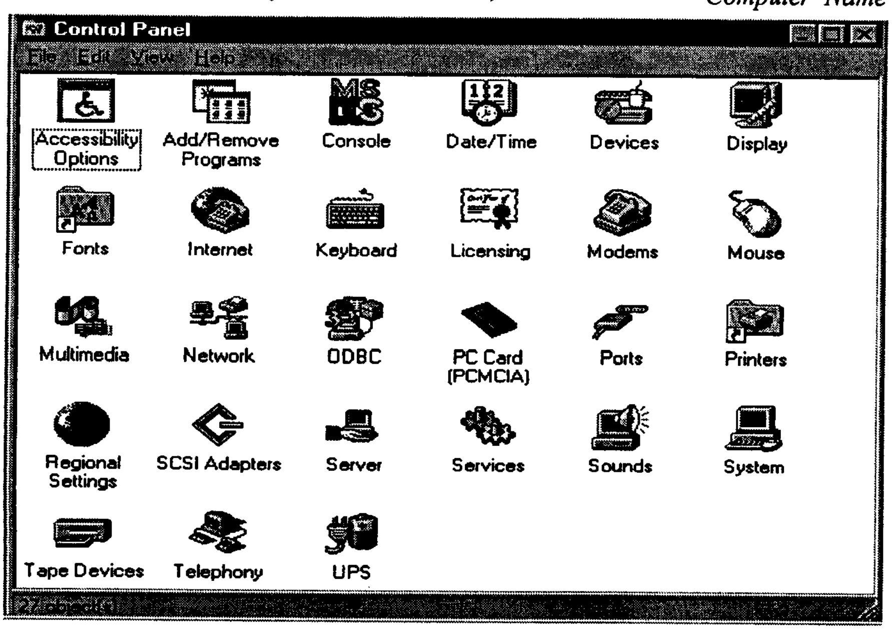


Figure 1. Control Panel

The server installation steps for a Primary Domain Controller (PDC) are outlined next. The initial step of the Windows NT Setup copied files from the CD to an area of the server hard disk; the setup program restarted the computer; displayed several screens filled with instructions, licensing agreement, etc.; the setup restarted computer again, and displayed the Windows NT Server Setup Wizard screen with three parts:

(1) *Gathering information about your computer.*
(2) *Installing Windows NT Networking.*
(3) *Finishing Setup.*

With the Wizard, all we had to do was to select a part, to complete the next dialog box, and to click the *Next* button in the dialog box. Administrators must know the implication of selecting an option and how to complete an open input box in the Wizards or the dialog boxes. Part One consists of the following seven dialog boxes: *Name and Organization, Registration, Licensing Modes* with two options - *Per Server* and *Per Seat* - which are two choices of Microsoft licensing schemes with instructions, *Computer Name* for the server, *Server Type* with three options - *Primary Domain Controller, Backup Domain Controller*, and *Stand-Alone Server - Administrator Account, and Emergency Repair Disk.* In Microsoft terms, *Domain* is a family of NT servers and workstations with a central security policy and user account database[5] maintained on the server - PDC. Each dialog box has *Back, Next,* and *Help* buttons. The first server being installed in a domain must be the primary domain controller of the domain. Part Two is the core of network administration including NIC setup, Internet Information Server setup, Gopher, FTP, and Remote Access Services setup. For the initial server installation, most of the network options were skipped, and part two was finished with the specification of *Computer Name* and *Domain.* Part three included selection of *Time Zone, Video Testing,* building of *Emergency Repair Disk* (ERD), and *Restart Computer.* After the emergency repair disk was removed from the computer, the server was ready to reboot.

We installed the network options using the *Network* setup icon in the *Control Panel* shown in Figure 1, after

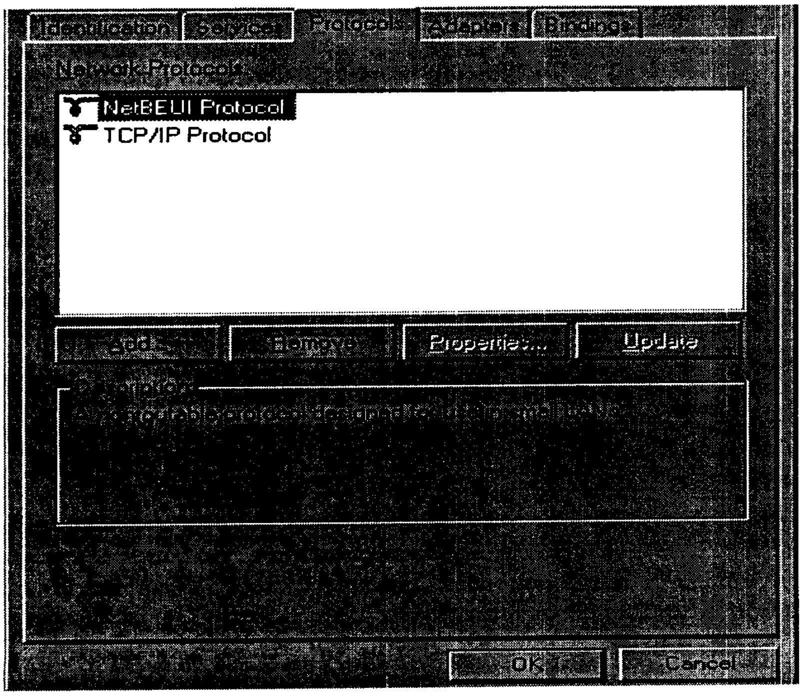


Figure 2. Network Setup Panel

setting up the server with minimum configuration. Across the top of the Network setup panel, five tabs were listed: *Identification, Services, Protocols, Adapters,* and *Bindings* as in Figure 2. Clicking a tab moved the panel associated with the tab to the top. Each panel has numerous buttons such as *Add, Remove, Properties or Configure, Update, OK,* and *Cancel* for protocols, adapters, and services setup.

Attempting to make a Windows 95 PC a client of the prototype server was unsuccessful at first because the NIC of the server was not properly installed with the beta CD package. After the SMC card being replaced with a 3Com adapter card, 3C509B, the Windows 95 PC became a client of the Windows NT 4.0 beta prototype server. By checking very closely the Windows NT 4.0 Hardware Compatibility List, http: //www.microsoft .com/ntsserver/hcl/hclintro.htm, and a hot-link of the list, we found that the driver of the SMC EtherCard Elite16T Ultra had to be manually installed with a disk with files copied from an appropriate DRVLIB directory on the CD.

By the time the Spring semester started in January 1997, Windows NT release 4.0 CD was available for the class; twenty-six students were divided into six teams; five C-class IP addresses were assigned to every team. Each team completed a Primary Domain Controller setup with the Windows NT 4 server package. Other networking classes which were using only DOS features shared the lab facilities; therefore, the *boot-loader* of the servers enabled the students to choose MS-DOS or NT to boot the computer. Most client nodes were MS-DOS based, some were Windows 3.11 or Windows 95 stations depending on the hardware. If more computers were available for the lab, backup domain controllers would have been added to a few domains. A workstation node software, *MS Network Client v3.0 for MS-DOS and Windows,* was included in the Windows NT Server 4.0 software CD; the client software was copied to two disks from the CD. Each team setup successfully MS-DOS client nodes with the disks. Every PDC supported DHCP and WINS. Next tasks were system administration functions: the creation of accounts for users, the creation of groups, the creation of shared directories, and the creation of print queues with Administrative Wizards. At the

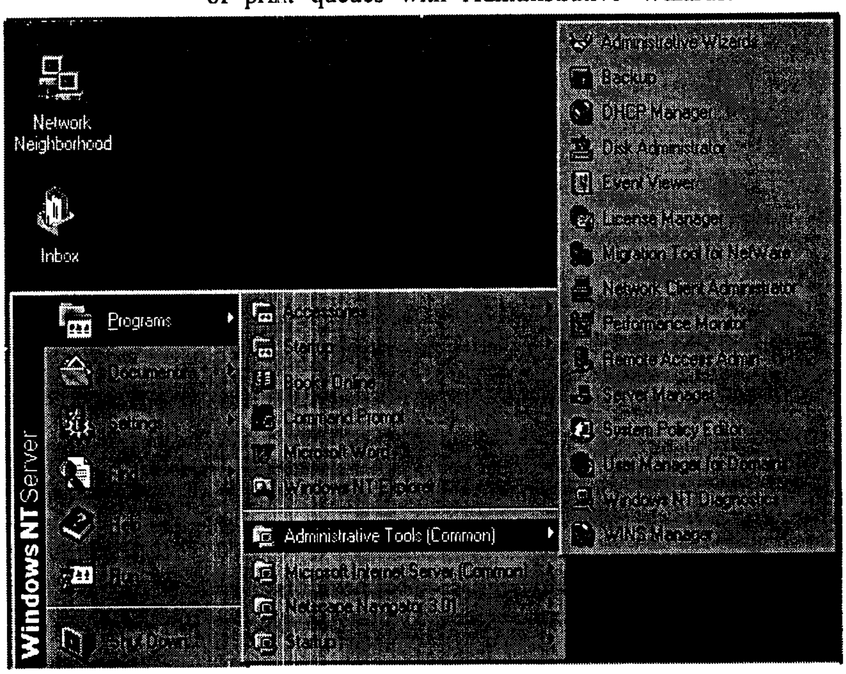


Figure 3. Administrative Tools

completion of the PDCs setup, ERDs were updated with the Repair Disk Utility, RDISK.

In Windows NT, an *event* is "any significant occurrence in the system or in an application" about which users should be notified.[4] Events are listed in one of three logs: *system log*, *application log*, and *security log*. To display entries of the system log, click *Event Viewer* of Administrative Tools shown in Figure 3. It has four menus, *Log*, *View*, *Options*, and *Help*. With event Viewer, we can filter or sort the log entries, and find details of a selected event by double clicking it. The filter dialog box in Figure 4 shows that events are divided into five different types: *Information*, *Warning*, *Error*, *Success Audit*, and *Failure Audit;* by checking type Error and unchecking other record types, only error records are selected. All errors that occurred during the server installation steps were recorded in the system log. On completion of Primary Domain Controller setup, the user accounts were created with another administrative tool, *User Manager for Domains*, which was available when the Domain Controller was setup successfully including a network protocol. User Manager for Domains, shown in Figure 5, is the primary administration tool and

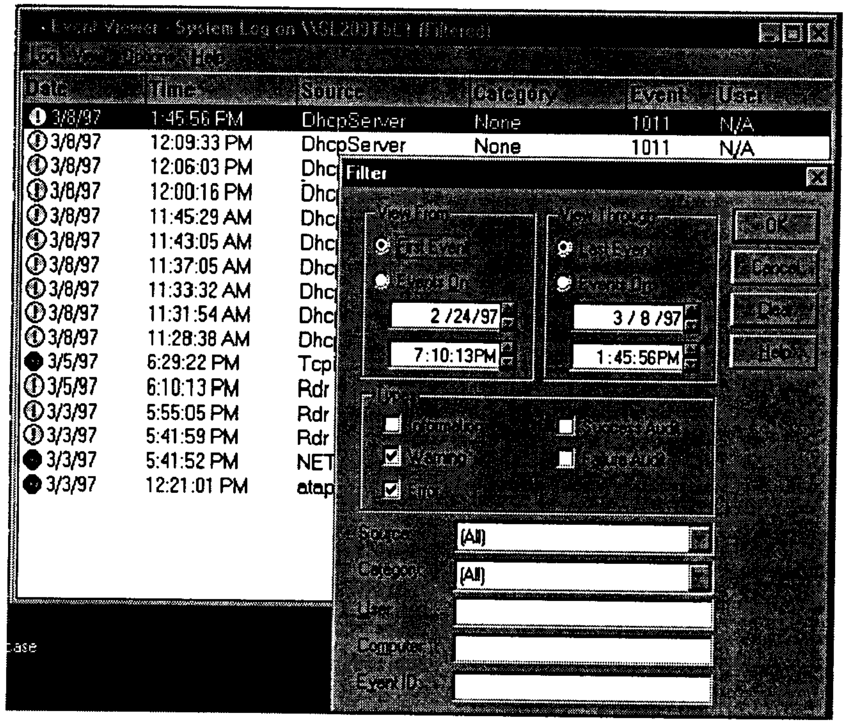


Figure 4. Event Viewer

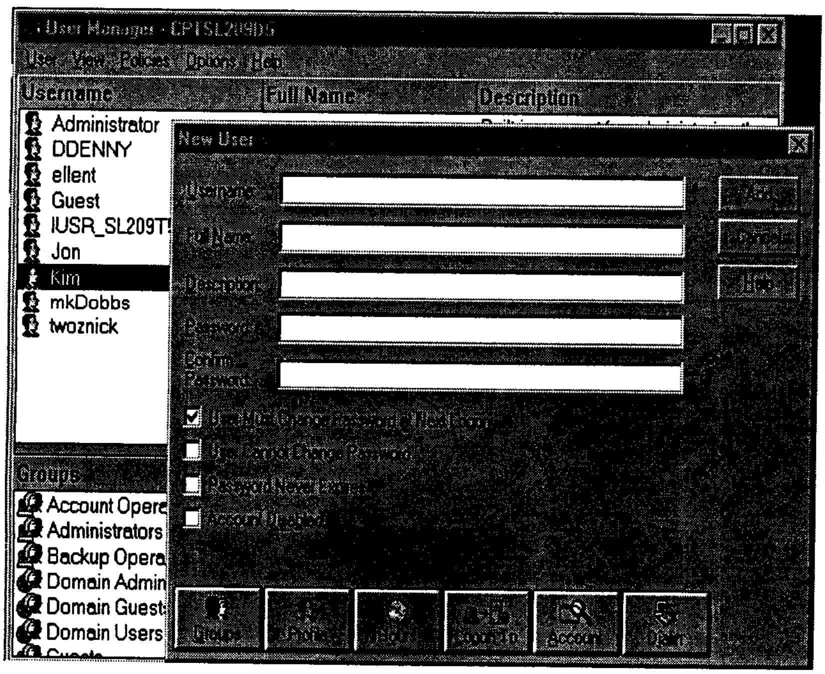


Figure 5. User Manager

has many functions for supporting a broad spectrum of users in an organization. It has five menus: *User*, *View*, *Policies*, *Options*, and *Help*. *Administrator* and *Guest* are two built-in users. Each domain supported a minimum of five to seven user accounts. A user can be a member of a group. *Administrators*, *Backup operators*, and *Everyone* are three of many built-in groups. To simplify users and group administration of multiple domains, Windows NT has two categories of groups: *local group* and *global group*. Details on the subject are outside the scope of the paper.

5. Shared Multi-Domain Network Facilities

Each team configured a single LAN NT domain with a PDC and one to three client workstations. Every PDC used per seat licensing mode. The lab LAN consisted of six domains. Domains are an administrative units as well as building blocks of a global enterprise network. Windows NT has a powerful, convenient feature called *Trust Relationships*[12] - a link from a *trusting* domain to a *trusted* domain. The administrators of two domains must reach an agreement on the link of their domains; they specify their domain names in the *Trust Relationships* dialog box available from the policies menu of the User Manger for Domains. Then, global groups of the trusting domain can be included in the local groups of the trusted domain. Now, members of the global groups of a trusting domain can use the resources of the trusted domain. Trust relationships were established among the six domains;

each team updated its ERD again. Close to the end of the semester, every user was able to use computing resources in other domains through trust relationships among domains.

6. Utilities, Auditing, and Monitoring

Among numerous topics not covered so far, only four topics were selected for the students to explore: They were: Performance Monitor, Windows NT Diagnostics, Task Manger, and Auditing. *Performance Monitor* and *Window NT Diagnostics* are two of fifteen common NT administrative tools listed in Figure 3.

Performance Monitor dialog box has the usual pull-down menu of *File*, *Edit*, *View*, *Options*, and *Help*; a tool bar for *View a chart*, *View the Alert*, *Add counter*, *Options*, etc. buttons; and a large window for line graph display.

The Windows NT Diagnostics has two menus, *File* and *Help*, and nine tabs including *Services*, *Resources*, *Network*, *Devices*, *Memory*, etc. The Services panel displays status, *running* or *stopped*, of every service including DHCP client, and Remote Procedure Call Service. By double clicking a service, more detail on the service can be displayed. All these Wizards and dialog boxes have on-line help facilities.

The Task Manager has menus and three tabs - *Applications*, *Processes*, and *Performance*. The processes tab lists processes, and has *End Process* button. The performance tab shows CPU and MEM usage. The Policies menu of the User Manager for Domains has an entry, *Audit*; selecting Audit displays the *Audit Policy* dialog box with seven events such as *Logon and Logoff* and *File and Object Access* each with *Success* and *Failure* selections. Any of these events can be enabled for auditing; audited events will be listed in the security log.

7. Conclusion

Windows NT is a comprehensive, evolving NOS with abundant services and utilities with consistent graphical user interfaces. The hands-on LAN lab course covered several services and utilities carefully selected. Skills for installing and supporting a service or utility are transferred to those for other services or utility. Students found the scheduled class time was not long enough to complete the assigned projects; every team worked in the lab during two to three week-ends. Every team included summaries of the RFC the team reviewed in the final project report. After mastering skills for configuring and supporting the carefully selected NT capabilities, students would be prepared to setup, manage, and support multi-operating system enterprise networks using on-line, context sensitive help facilities, and to understand imminent technologies such as the emerging IP standard, IPv6.

REFERENCES

1. Y.J. Kim and E. Solinski, "Renewal of Network Systems Curriculum with Emphasis on Students' Lab Projects, " ISECON '95 Proceedings, November 3-5, 1995, pp. 125-132.
2. Information Systems: IS'95 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, Draft Report, May 1995, AIS, ACM, DPMA, p. 66.
3. T. Lewis, "The Next $10,000_2$ Years: Part I," *Computer,* April 1996, pp. 64-70.
4. M. Minasi, C. Anderson, and E. Creegan, *Mastering Windows NT Server 4,* Third Edition, Alameda, CA., Network Press, 1996.
5. K Siyan, *Windows NT Server Professional Reference*, Indianapolis: New Riders Publishing, 1965, p.164.
6. D. Heywood, *Inside Windows NT Server 4*, Indianapolis: New Riders Publishing, 1966.
7. J. Grams, et al., *Windows NT 4 Server UNLEASHED*, Indianapolis: SAMS Publishing, 1996.
8. Release Notes for Version 4.0, Microsoft Windows NT Workstation and Microsoft Windows NT Server, Microsoft Corporation, 1996.
9. D. M. Sutton, Installation of Windows NT Server 4.0 Beta, CPT240 term paper, July 1996.
10. W. Stallings, *Data and Computer Communications*, Fifth Edition, Prentice Hall, 1997.
11. K Siyan, *NetWare Professional Reference*, Fourth Edition., Indianapolis: New Riders Publishing, 1965.
12. Domain Planning Guide, Microsoft document downloaded from http://www.micorsoft.com/ ntserver.
13. http://playground.sun.com/pub/ipng/html/specs/specifications.html in http://playground.sun.com/ pub/ipng/html/ipng-main.html

A Total Educational Experience Through Cooperative Partnerships
(A Successful Internship Model for CIS Programs in the 1990's)

Sharon N. Vest, Marino J. Niccolai,[1]
Ray Alstadt[2]

ABSTRACT

A pilot study and a follow-on full scale internship agreement, between academic degree programs in computer and information sciences (CIS) and an local, industrial partner, were developed and successfully deployed. This article describes the mechanics of establishing and managing a unique internship program, the barriers and benefits to the industrial partners, to the computing disciplines, and to the students, and provides lessons learned from two years of experiences.

The organization and careful management of an academically based internship program, by the CIS faculty, has lead to significant improvement in communications between students and faculty, and between the faculty and local industrial representatives. The expanded role of the faculty now includes the role of industrial mentor as well as the more traditional "lecture-advisor" role and, not surprisingly has led to a greater appreciation of academic disciplines by the students.

INTRODUCTION

During the past two and one-half years, the School of Computer and Information Sciences , has been involved in an extensive, comprehensive experiment with a 'bleeding-edge' technology company to jointly develop and administer a parallel internship program. The first year of the program was considered a pilot study and was administered in an ad-hoc, as needed manner with no resources allocated to manage the program. Though an administrative nightmare, it was a great success! The industrial partner was extremely pleased with the quality of support, students received valuable experience and compensation, and faculty noted increased academic motivation among participants. Recent graduates who participated in the internship reported that the experience provided a strategic advantage during recruitment.

Lessons learned from the first year resulted in significant changes that greatly improved the quality of faculty-students-project supervisor interactions. This article describes two different approaches to supporting a comprehensive internship program (the first is our pilot study model and the second is the full-scale implementation). The authors provide insights into establishing relationships with industrial partners to create internship agreements that support student projects (to directly benefit the industry) , as well as faculty research and scholarship opportunities.

The faculty of the School of CIS created a unique and innovative solution to the problem of providing relevant work experiences while providing a full-time quality academic program. The authors believe that our experiences can and should be translated to other computing and information sciences programs. **Our solution was to establish a large, funded grant from a technology partner to the School of CIS that is administered by the Principal Investigators (PI) and is used to employ up to thirty (30) advanced undergraduate and graduate students in a parallel internship program intended to provide them with an integrated educational experience.**

BACKGROUND

Many CIS students attempt to supplement their educational experiences with practical skills that are related to current employment demands. While this may be a desirable goal, an unstructured approach is counterproductive since students may become skilled at low level programming tasks, seek permanent employment and forget their primary goal of completing a degree program. Thus, there exists a need to help define meaningful and academically sound programs that provide "real-world" experiences while keeping students motivated to complete their degrees. At the same time, most professional agencies support the need for more interaction between academic programs and industrial sponsors. To achieve both goals, CIS departments can develop strong internship

[1] Niccolai and Vest, University of South Alabama, School of Computer and Information Sciences.
[2] Alstadt, QMS, Inc.

opportunities with local industries. Our experiences show that these internships must be carefully managed to satisfy high quality standards.

The internship program that was established and is described in this paper addresses these major concerns by targeting a local technology company that was willing to partner with the university to involve a large number of interns in their software development processes. The challenge was to create a funding mechanism by which internship projects could be identified by the industrial partner and students could be recruited from the university, interviewed and employed on a long term basis while maintaining normal course loads and reasonable GPAs.

YEAR ONE: THE PILOT STUDY

For several years prior to the start of a formal internship agreement, a CIS faculty member had received small, continuing grants from a local 'high-tech' firm to provide project and process support. Each of the grants involved one or more student assistants who received compensation. In some cases the students were employed to provide programming assistance and in other cases they worked in teams as part of their senior project classes. In the former cases, student assistant salaries were written in as part of the grant and they were paid, but received no formal academic recognition. In the latter cases, the student assistants were not paid, but did receive academic credit for their contributions.

Everyone from the company and the university agreed that the work produced was excellent. The students were satisfied that they had received valuable, practical and relevant experience. Therefore, a contractual mechanism was sought: to formalize these arrangements, to expand the scope to include more part-time student assistants, and to form a partnership to provide a total educational experience for the participants. This experience involved integrating classroom theory and understanding with the realities of on-the-job skills and requirements. At the same time, all of the students would be paid and receive the academic recognition they deserved.

After several false starts, the idea of providing a pool of funds, in the form of a contract or grant, to the university from the local industry was decided upon. The funds could be used to contract for assistance on specific projects or could be used to provide on-the-job experience for the student interns. The grant proposal that was developed, for the first year, involved ten to fifteen internship opportunities. The duration of each was to range from one academic quarter to one year. Each intern was expected to work up to, but not more than, twenty (20) hours per week while maintaining a reasonable academic load. The proposal included funds to support the interns at salaries in the range of $7.00 to $8.00 per hour. A small, additional university overhead charge was included. The grant was signed and a number of hand-picked interns were selected to work with supervisors on different, relevant projects.

What happened next was beyond anyone's expectations. It was thought that the interns would require a long training curve before they could be "trusted" to work on their own projects. **WRONG !** The interns were well prepared and so excited about the opportunity that they surprised both the faculty and the project supervisors with their abilities and skills. In just a few months the interns were contributing at a far higher level than expected and they began to do better in their studies at the university as well. The PI for the pilot study grant was overwhelmed with request from other students to join the internship program.

As long as less than ten interns were involved, the internship program progressed on its own with little administration required. However due to the unexpected success of the program, more and more software development managers requested internship support. Soon, every industrial manager and every project team was **demanding** that they be allowed to participate in the program. Student interest (and apparent academic benefit) was obvious and the demand for their services was pressing. In an attempt to satisfy both demands, the program was allowed to grow and grow and grow! In less than a year there were over forty (40) interns each working up to twenty (20) hours per week. They soon became an indispensable part of the company's work force. Obviously the budget was shot and had to be re-negotiated several times. It became increasingly more difficult to check on academic performance and as a consequence some grades slipped. To satisfy such a large demand for interns and in an attempt not to artificially slow the growth of the program, students with less than ideal GPAs were allowed to participate. However, the biggest concern was that there was no time to properly advice, counsel, or manage the individual interns relative to their academic programs.

The first year's pilot program ended on a high note. The industrial partner was excited and completely satisfied with the interns' contributions. The students were happy, paid and receiving relevant work experience. The university received a large grant ($500,000) enabling it to employ the students and to provide meaningful external educational experiences under the guidance of the CIS faculty. This was a WIN, WIN, WIN situation.

The obvious question was: Could the internship continue to be successful while managed in the ad hoc manner used during the pilot study year?

YEAR TWO: A MORE MANAGEABLE INTERNSHIP MODEL

The rapid growth of the program in the pilot year clearly indicated that a real need of the university and of industry was being addressed. Industry had available a quality source of manpower, the university received external funding and the School of Computer and Information Sciences was able to add an application dimension to classroom theory.

Because of the program's unforeseen growth , sufficient resources had not been allocated for administration. The first year experience resulted in a loss of control. The budget had to be appended several times during the first year due to a much higher than expected demand for interns. Each new contract had to be renegotiated and caused a confusing array of payroll accounts to be created. An academic audit at the end of the year revealed a few students were given intern positions while not adequately prepared academically. Some interns worked more than the 20-hour university maximum. A close international student community produced a disproportionately high number of international student interns as compared to other groups. The need to advertise the program school-wide and to provide this opportunity to all students was apparent. Finally, the loss of some students to full-time employment was directly attributable to their internship involvement.

The principal investigator, the School's chair and the Dean recognized the benefits of the program, but were very concerned about the legal and management issues relevant to a university-sponsored program that allowed students to work off campus. The first step in incorporating the program as a permanent option in the academic program was to expand the resources by an additional one-third time faculty allocation. The program had to be clearly defined. It was a program to enhance the educational experience of the students. It could not be used as an employment agency. The industry's demand for a maximum number of interns working up to 40 hours a week had to be balanced with the educational demands of meeting degree requirements. The program had to be managed so that students could reach their academic goal—a bachelor's or master's degree in Computer and Information Science. If students were allowed to work in excess of a part-time basis, or could be coaxed into accepting a full-time position while deferring their education, the program would be viewed as a failure. Concerns that had to addressed were: legal issues associated with employing students, unfair competition with local employment agencies, and intern safety while on the job.

Once the decision to continue the program was made, the principal investigators called two meetings, one with the industry managers and one with the interns. The initial success of the program and the ensuing concerns were discussed. Emphasis was placed on the goals of the program: **to enhance the students' educational experience; and to meet the need of industry for qualified personnel.** The need for more controls was noted as requisite to meeting these goals. Both groups were assured that their suggestions would be considered in the continued management of the program.

To avoid underestimated financial need, the contract was stated in terms of a maximum dollar amount. The principal investigator was charged with the responsibility of quarterly reports to insure that the program remained within the prorated budget. The industry managers would be required to limit their requests for number of interns and hours worked based on the annual agreement.

The entire program was examined to find areas of improvement from the application process to intern termination. Clear policies and procedures were developed to manage the program. These additional controls led to a paper trail of program documentation.

Obviously, accepting a struggling student into the program that required additional work away from studies would be counterproductive. The disproportionately high learning curve of a weak student would drain resources without adding value for industry management. To adequately screen students, an application process was developed. The students were required to complete an *Internship Application Form* that clearly stated the minimum qualifications: sophomore standing; completion of sophomore programming sequence; appropriate residency status; and minimum grade point average. Students were required to submit a resume with the application and the school secretary attaches a copy of the transcript. Each applicant was given an *Internship Guidelines* handout that clearly states the program's policies and procedures.

Industry managers now must complete an *Intern Request Form* for each new position. This procedure replaces the emergency calls from managers relaying a project deadline that requires an intern or two ASAP. The new request procedure encourages managers to plan personnel needs rather than use the crisis management technique of the first year. The request form provides details of the position, including prerequisite skills and a realistic job description. This information is used by investigators to select appropriate student applicants for interviews. Two or three students are selected for interviews with the requesting manager. The manager notifies the investigators of his choice and the acceptance process begins. Typical job descriptions include:

⇒ Network wiring, network traffic monitoring, cc:Mail, script writing, NT and win95 installation.

⇒ Component testing.

⇒ Development of product advertisements.

⇒ New PC system setups, repairs and upgrades.

⇒ PC/UNIX application and driver testing and/or print engine testing..

⇒ Test printers—performance and color quality.

Product testing including but not limited to printers, host software, network and daughterboards for design defects.

Once a student is accepted into the program, s/he must sign an *Internship Agreement* that details program policies. Each new intern is given a copy of the University's *Student Employee Personnel Policy* handout outlining the university's policies and procedures for student employment. This handout addresses such issues as on-the-job injury and workloads during and between academic quarters, as well as relevant definitions such as *full* and *part time student*. The agreement specifies the maximum number of hours an intern can work, and states the academic criteria to remain in the program. In an effort to avoid any future confusion when issues of increased desire to work by interns or management arise, the intern is clearly informed that the goal of the program is to enhance his educational experience, not his pocketbook. Any rulings on future conflicts between the two opposing objectives will always favor the educational one.

An orientation for all new interns is held at the beginning of each quarter to communicate the program goals and expectations. The interns are required to register for one hour internship credit each quarter. They will receive a grade of *S* (satisfactory) or *U* (unsatisfactory). This becomes part of their academic record at the university and may be used to satisfy some degree requirements. Each intern must submit an approved *Work Study Plan* the beginning of each quarter. Interns are given a copy of the *Intern Evaluation Form* used by management for evaluation of their work performance each quarter. New interns are provided e-mail addresses of the investigators and reciprocate by providing theirs to insure timely responses to all communication.

One meeting per quarter is scheduled to disseminate information to all interns and update the status of the program based on changes implemented during the year. This meeting provides a forum for interns, management and investigators to provide input into the program. E-mail has turned out to be the most effective way to communicate among 40 plus interns with varying class and work schedules and respective managers.

At the end of each quarter interns with management evaluations that identify problem areas, or grades that fall below standards are scheduled for a conference. A plan for improvement is developed for the following quarter. If improvement is not made, the intern is removed from the program with the opportunity to reapply after one quarter, provided academic standards are met.

A file is maintained on all interns that includes application, resume, grade reports, evaluations and work study plans. When an intern is terminated, a *Termination Report* is completed. Reasons for leaving the program include graduation, project completion, academic probation, poor work evaluations and personal choice.

Quarterly, a *Budget Summary* is prepared illustrating total hours worked and proportion of grant funds used to date. This report is submitted to the dean, chair and vice president of the industry partner.

WIN, WIN, WIN: PROS (and a few CONS)

Pros

All participants, the industry partner, the university and the students are progressing through the program as winners. The industry partner has access to a well-prepared personnel resource to meet their project demands on an "as needed" basis. In an extremely competitive market, this resource enables industry to meet fluctuating labor demands without long term commitment. Management also has a very effective screening process in place to choose future full-time employees. The added public relations advantage of industry cooperatively supporting a community educational institution is also noted as a plus to the program.

The university receives revenue in the form of faculty release time compensation and overhead. The program has proved to be an effective recruiting bonus in attracting prospective students. The intangible benefits derived from a positive relationship with local industry and the community should not be overlooked.

The faculty have learned that working with industry can be beneficial to the academic program. A new degree of student concern about class performance related to program eligibility has been noted. Synergy among the faculty has generated publication ideas and additional suggestions for program refinement and expansion.

The students are given an opportunity to sample typical job assignments in a technology-based company, and to apply classroom theory. The program presents an added incentive to maintain a high level of academic performance. Feedback from graduating seniors is overwhelmingly positive, and indicates that the intern experience is a major point of interest during interviews for full time employment. E-mails from former interns state the internship involvement gave them a competitive edge over other applicants.

Cons

The internship program does present constraints to management not imposed by employment agencies relative to supplying skilled part-time personnel Management is required to keep hours worked per intern within university guidelines, and occasionally loses a very productive worker because grades fall below the academic standard.

The university bears an additional liability burden associated with student employment off campus and apparent competition with employment agencies. Faculty in the technical sciences are scarce. The program consumes one-third of one faculty's allocation previously used to meet the school's instructional needs.

Managing the program presents a significant work load for the faculty. Maintaining the budget, monitoring student performance on the job and in the classroom, and meeting the labor demands of the industry partner while maintaining an educational focus are very demanding tasks.

Students can be lured by the immediate appeal of increased income. Good workers have been recruited by the industry partner to work full time, risking the timely completion of their educational goal. The investigators discuss regularly with both management and interns the very real concern for the potential long term damage to the student's personal and professional goals. Currently, management strongly supports the goal of graduation prior to employment.

CONCLUSIONS

The second year is proving to be extremely successful. Plans for the future include specific data collection on the progress of the interns to investigate what impact, if any, the program has on their careers. Exit interviews are being developed with subsequent surveys at 1 and 5 year intervals after graduation. Expansion of the program to other industry sites is another goal. Expansion would require a minimum of one full time faculty and one-half staff resource allocation.

Currently feedback from all involved is positive. The interns appreciate the opportunity to work in a related field and are progressing toward graduation. The industry partner is very pleased with the caliber of personnel provided. The university is pleased with the additional funding and the enhanced relationship with local industry and the community and, reassured by the new controls implemented.

The internship model presented in this paper, demonstrates how the total educational experience can be expanded through cooperative partnerships with industry. The model provides a "win, win, win" solution for all participants. The student gain relevant experience; industry become involved in educating and training future generations of employees; the university solidifies its role as an equal partner in the community. Because the program is managed by the CIS faculty, it insures high quality and can be ported to other CIS programs where appropriate technical partners can be identified (this constraint is easily satisfied by most institutions). Plans for continued success in year three will include repetition of the process loop of evaluation and feedback to improve the program quality and better meet the objectives of all participants.

Rethinking Computer Architecture in the Undergraduate IS Curriculum: A Cognitive-Based Approach

Irv Englander, PhD.
Bentley College
175 Forest Street, Waltham MA 02154
e-mail: ienglander@bentley.edu

Abstract

Undergraduate-level computer architecture has traditionally been one of the most difficult and frustrating courses within the IS discipline, both for the students and for the instructor. A cognitive-based approach to this course, based on a model developed by Doran and Langan, has been developed and tested in our curriculum. Initial returns indicate that the model successfully motivates and excites students, resulting in a high degree of learning as measured by examination scores and quality of projects, low drop rates, and high overall satisfaction with the course

Overview

The Bentley College CIS curriculum has included a required course in basic computer architecture in the Master's degree program since 1983. A similar course was offered at the undergraduate level from 1983 to 1988, but was discontinued at that time because the course did not fit into the structure of the revised curriculum introduced in that year. In 1996, another major redesign of the undergraduate curriculum resulted in reintroduction of a one semester computer architecture course, CS 220, as a core requirement of the curriculum. [1]

During the intervening period, 1988-1996, a number of changes took place that necessitated development of CS 220 as an entirely new course design. The most obvious of those changes is the continuing explosion of technological change. This factor places an emphasis on the need for a knowledge base and problem-solving skills upon which a future IS professional will be able to evaluate and understand new developments as they occur.

Other factors, more subtle, but of equal importance, are the changing nature of the student body entering college and the increasing, but uneven, computer sophistication of the users as they enter the undergraduate CIS program. Our recent experience indicates that many of today's entering college students have less natural motivation towards studying, shorter attention spans, and a tendency towards memorization of facts in place of problem solving skills than did previous generations of college students. More effort is required to excite and challenge these students. There is an increased need to show immediate and long-term relevance to motivate a student to put maximum effort into a course. These traits are unfortunately combined with a need to work longer hours outside to pay for college.

At the same time, the range of experience of students entering the CIS program has never been broader. The content, level and quality of teaching in computer courses at the high school level is highly variable, ranging from no coverage to (often poor) language-based courses to in-depth coverage of computer concepts. In many cases, high school computer courses have alienated students and reduced interest in IS technology.

Finally, the end goals and motivation levels of undergraduates in CIS tend to be substantially different from those of students graduating from the MSCIS program, thus it is not appropriate to "water down" the graduate level course. Although there is some overlap between the courses, the undergraduate course calls for different coverage and the use of different teaching techniques in each course.

The charge of the undergraduate computer architecture course at Bentley College is to present the basic concepts of computer hardware, software, data representation, and the technical aspects of data communication, with an emphasis on fundamentals that can be used by students to maintain their technical currency on their own. The course is normally taken early in the program, and serves as a prerequisite to an in-depth course in data communications. The course specification is consistent with IS-4 in the 1997 Joint Undergraduate Information Systems curriculum. [2]. Currently, there are a total of 150 students enrolled in four sections, taught by two faculty members.

Pedagogy

A number of studies have focused on the challenge of teaching difficult technical concepts using a cognitive-based approach, particularly in the areas of computer programming and introductory computer science. Especially noteworthy is the curriculum development project by Doran and Langan [3]. This work was augmented by an NSF Workshop at the University of South

Alabama in 1996 on faculty enhancement using a cognitive-based approach to the teaching of introductory computer science courses, which was attended by the author of this paper [4].

In their paper, Doran and Langan note that initial Computer Science courses are often characterized by student frustration, a fairly significant drop rate, poorly defined exit skills, and inadequate tools for measuring student learning. In response, they developed a model based on cognitive learning. Specific features of their work include

- strategic sequencing and associated levels of mastery based on Bloom's [5] taxonomy of learning. Briefly, Bloom defines learning in terms of six levels, ranging from factual knowledge (level 1), through comprehension, application, analysis, synthesis, and judgment.

- course documents that explicitly define and schedule course micro-objectives, that map each micro-objective to a specific Bloom knowledge level, and that help to achieve and measure success

- the appropriate use of prerequisite knowledge [6], in which the successful completion of each topic lays a foundation for subsequent topics, at the micro-objective level

- a spiral approach to presentation

- frequent feedback

- integral use of structured closed labs

The model allows faculty management of frustration and motivation, provides clear goals and means for reaching those goals, offers clear measurements of student achievement and success, provides reinforcement of good study habits, and invites student involvement in their own learning. In fact, student success and awareness of their own success is a central focus of the entire design.

We observed that the course characterizations described by Doran and Langan were consistent with our own past experience in the undergraduate computer architecture course, and with the motivational factors to be addressed. In addition, the model addresses clearly the issues of motivation, reinforcement of good study habits, and active student involvement in the learning process. Although there are differences in specific components (e.g. the use of structured closed labs is not applicable to the computer architecture course), it appeared that many of the ideas presented in their work would have direct relevance to us. This suggested that a cognitive-based approach would be appropriate for the new CS220 course.

Therefore, we have adapted this model to the computer architecture course. The salient features of our course design are:

- A modular framework, in which each module establishes clearly laid out goals and expectations, and a method to reach the goals. Explicit within this framework is the use of Bloom's taxonomy as a means to measure learning objectives and achievement. Conscious awareness of Bloom levels of achievement is reinforced continually throughout the semester.

- A strategic linear sequence that emphasizes use of successfully completed objectives as a foundation for subsequent topics

- A careful balance of theoretical base material with relevant current application of the material

- A lecture/textbook format, heavily supplemented by classroom and outside activities designed to continually reinforce the importance and relevance of the material to the student. The outside activities that we have created are designed to maintain student interest and involvement in their own learning and to validate student understanding of the classroom material

- Maintenance of student journals, as a means for immediate student reinforcement of learning. The journals also serve as a metric for student accomplishment and success.

Proactive success motivation is addressed through the use of goals that are sufficiently small and clear to assure success, through the use of a grading system that minimizes judgmental evaluation, described below, and through the student journals.

Specifics of Implementation

Our implementation of the cognitive-based model includes the following components, tools, and features:

- **Lectures and textbook reading assignments**. The purpose of the lecture/textbook component is to bring students of diverse backgrounds to a Bloom level 1 on the learning scale. Lectures and textbook reading also provide guidance for students to approach Bloom level 2. The classroom is also used for a number of exercises designed to show students practical examples of the technology discussed in class. RealAudio and

Quicktime video demonstrations allowed students to see the application of data formatting in use. A class period devoted to the actual upgrade of a computer by replacement of a PC motherboard moved discussion of the various components of the computer into a real world experience. (The latter class proved to be particularly successful as a motivating tool. Many students have commented that this class connected together many of the ideas discussed in an exciting and positive way.)

- **Detailed Course Modules**. Each area of learning is supported by a detailed course module that provides specific goals, learning objectives, and topics of coverage, with specific typical questions for each topic. Bloom levels are assigned to each micro-objective and identified for each typical question. Doran and Langdon also provide glossaries of the terms being used, but this was not necessary in our case, because the textbook provides appropriate glossaries at the end of each chapter.

  Eight modules comprise the course coverage for CS 220. These cover number systems and data formats (6 hrs.) the CPU and memory (6 hrs.), input/ouput (3 hrs.), modern hardware system concepts (6 hrs.), operating systems (6 hrs.), file systems (1.5 hrs.), assembly and compilation (1.5 hrs.), and data communication and network technology (6 hrs.). An abridged version of the CPU/memory module is shown in Appendix 1 to this paper.

- **Homework Problems and Miniprojects**. There are homework assignments and/or miniprojects associated with each course module. These assignments serve as practice, with the specific aim to reinforce understanding at the appropriate Bloom level. Miniprojects also show practical modern application of the topic and promote a feeling of understanding by allowing the student to recognize his/her ability to read and interpret the literature successfully. Assignments and miniprojects are graded on an outstanding/ pass/not-accepted basis. The student may resubmit a not-accepted effort for a passing grade on the assignment. Therefore students are encouraged to succeed.

  In particular, the miniprojects are designed to use the current technological explosion to create an atmosphere of excitement and involvement. We have observed that students generally enjoy Web surfing, so the assignments reflect that interest. Our specific miniproject topics for 1996-1997 included:

  - ♦ A 3-4 page paper that uses information on the Web to describe a specific multimedia data format or to compare the apecifics and performance of two or more data formats.

  - ♦ A 3-4 page paper that uses information on the Web to describe differences between Intel MMX technology and standard Pentium technology.

  - ♦ A miniproject to use information on the Web to compare two different operating systems. Suggestions included comparing MacOS with the Next and BeOS systems, or comparing Windows 95 and Windows NT, or comparing Xwindows-based Unix with Windows 95. The handout for this miniproject is shown in appendix 2.

- **Journals**. Students are expected to provide a brief entry in a personal course journal for each class that they attend. The journal entry is to be made later in the same day as the class, and is to include one or more ideas learned during class that they consider important, relevant, and/or particularly interesting. The journals encourage the students to revisit the material while it is still fresh in their minds. They also allow a student to observe her/his own progress. In addition, students submit their journals to the faculty several times during the semester, and provide a means for the faculty to measure success and identify potential problems.

- **Exams**. Two exams are given during the semester. Students enter the exams knowing what to expect, since the exams reflect closely the coverage indicated in module handouts. The exams are designed to test knowledge at the Bloom levels indicated for each topic.The Bloom level for each question is indicated on the exam sheet so that the student clearly knows what level of understanding is expected in the answer.

To test the model, applicable course materials were developed and used for the graduate level computer architecture class in the semester prior to their introduction in the undergraduate course. This was not entirely possible, since the course content and student motivation are somewhat different, but the results indicated probable success. Even at the graduate level it was possible to observe improved understanding and excitement. As a result of the experiment, the materials have been fully deployed in the CS 220 course.

Interim Results and Conclusions

One of the observations made during graduate testing of the model was that teaching within a cognitive-based framework requires a substantial amount of preparation.

One of the questions that arose during the design of this course was whether the expected results were worth the additional effort required.

To date, our measures strongly indicate that the cognitive-based methodology represents a successful approach to this material, and is worth the additional effort. Exam grades have been generally high. The course drop rate is extremely low. In one section of 40 students, only a single student dropped the course. Students attend class regularly, and generally provide good explanations when they have missed class or will miss class, even though class attendance is not enforced. Homework was generally turned in on time. The number of "resubmits" required started at between 10-15%, and declined as the semester progressed. The quality of work on miniprojects is high.

Although early comments in the journals noted considerable initial frustration, particularly towards the math component, later journal entries indicate that the students feel a sense of accomplishment and a clear sense of their understanding of the course material, together with a high degree of interest and satisfaction with the course, even though many of the students continue to find some of the material challenging or difficult.

As an interesting sidenote, a number of students have expressed the opinion that the constant use and reinforcement of the Bloom taxonomy in CS 220 improved their overall understanding of the learning process, and provided a means for them to measure their learning and achievement in other courses.

As a result of our success this semester, we plan to continue development of the material to support cognitive-based learning in CS 220. In particular, we will expand the detail level of our module handouts for the next pass through the course and create new classroom teaching materials and exercises. We have concluded that a cognitive-based approach is highly suitable for the IS computer architecture class.

References

[1] Waguespack, L., Chand, D., Robertson, D., "Crafting a Business Bachelor of Science Program in Information Systems Following on IS'95", **Proceedings of the 13th Information Systems Education Conference** (ISECON), St. Louis, MO October 1996, pp. 7-12.

[2] Davis, G.B., Gorgone, J.T., Cougar, J.D., Feinstein, D.L., Longenecker, H.E., Jr., **IS '97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems**, ACM, AIS, and AITP, 1997

[3] Doran, M.V., Langan, D.D., "A Cognotive-Based Approach to Introductory Computer Science Courses: Lessons Learned", **Proceedings of the 26th SIGCSE Technical Symposium on Computer Science Education**, March 1995, Nashville, TN, pp. 218-222

[4] ---, **National Science Foundation Undergraduate Faculty Enhancement Workshop: A Cognitive-Based Approach to Implement the Introductory Computer Science Courses**, University of South Alabama, Mobile AL, June 9-16, 1996

[5] Bloom, B.S., et. al., **The Taxonomy of Educational Objectives: The Classification of the Education Goals. Handbook I: The Cognitive Domain**, McKay Press, New York 1956

[6] Gagne, R.M., Briggs, L.J., Wager, W.W., **The Conditions of Learning**, Holt Rinehart and Winston, New York 1988

Appendix 1
Abridged CPU/Memory Module

Reading: Textbook, chapters 6 and 7

Goals: To understand the operation, organization, and instruction set of a simple computer.

Overview of Module:

(1) Introduction to Little Man Computer (level 1).
(2) Concept and contents of a computer instruction set (level 1-2)
(3) Organization of real computer CPU (level 1); CPU registers (level 1); buses (level 1-2); memory (level 2)
(4) The CPU fetch-execute cycle (level 2-3)
(5) The use of the computer's instruction set to create simple programs (level 2-3)

Topics:

(1) The Little Man Computer organization, components, and instruction set; use of the instruction set to create simple programs
(2) Design of a simple CPU and correlation to components of the LMC
(3) Concept of registers in a CPU
(4) Organization and operation of random-access memory, usage of memory address and memory data registers
(4) Implementation of each LMC instruction, in terms of the step-by-step tasks performed by the LM. The use of registers in the CPU to implement the LMC instuctions; the Fetch/Execute instruction cycle. Equivalency of the tasks performed by the LM to the F-E cycle.
(5) Simple buses; the movement of data between different parts of the CPU

Proficiency:

You should be able to explain the organization of a CPU and the role of each component at a Bloom 2 knowledge level. This means that you should be able to explain the operation of a CPU in which the components differ slightly, but not in a fundamental way, from those described in the book.

You should be able to read and interpret simple programs written in LMC instruction code (level 2+). You should, with sufficient time and sufficient guidance, be able to write very simple programs in LMC code (level 2+ - 3).

You should be able to explain in detail the operation of an instruction for which you have been given the fetch-execute cycle (level 2). You should be able to create an F-E cycle for an instruction that is a slight variation on the ones studied, given the task that the instruction is to perform.(level 2+ - 3).

You should be able to discuss the components and characteristics of a bus (level 1-2-), and consider CPU performance in the context of a particular bus design (level 2+). You should be able to identify relevant tradeoffs in different bus designs (level 2).

Typical questions might be:

- Identify and explain the function of the important registers in a CPU (level 1).

- If I produce a CPU design which provides two parallel ALUs, what effect would that have on system performance compared to a single ALU design running at the same clock speed? (level 2). From what you know of the CPU instruction set, what are the limitations of such an approach? (level 2+ - 3)

- What are the different types of lines that would ve found on a bus? (level 1). If a particular bus multiplexes the data and address lines, what are the effects on system performance? Explain your answer. (level 2)

- Given a particular F-E cycle, different from those that you have already seen, create a table that shows the values stored in each relevant register at each step of the cycle, and explain what the instruction does. (level 2).

- Write a program in LMC code that produces the absolute value of the difference between two numbers. (level 3)

- Trace through a particular program, and state what it does in one sentence. (level 2 - 2+). What are the final values found in particular memory locations? (level 2) What are the final contents of particular registers? (level 2)

Also look through the exercises at the end of chapters 6 and 7 for specific examples.

Appendix 2
OS Miniproject

Due: 22 April 1997

Goal: To solidify your ability to use your knowledge of operating systems to evaluate and compare features and capabilities of different operating systems in a simulated real-world environment.

The task:

Your company is presently supporting the use of operating system "A" (see below) throughout the company, but the end users are clamoring to get ahold of the features of operating system "B". The CIO of the organization has requested your services to evaluate the technical features and advantages that would be obtained by making the change. For this purpose, you are to produce a report comparing the two operating systems, in terms of features, capabilities,user interface, internal differences, etc. Only operating system issues are to be considered (i.e. no hardware changes that might be required for the conversion.)You are limited to four single spaced types pages for this report.

You may choose any of the following comparisons ("A" vs. "B") for your project. Other comparisons may also be acceptable to the instructor, but check with the instructor first if you have another preference. At least one of the two OS's selected for comparison must not exist as a detailed example in chapter 18 of your textbook.

- Windows 3.1 vs. Windows 95
- Windows 95 vs. Windows NT
- Windows 95 vs. Windows 97
- Windows 95 vs. MacOS
- MacOS vs. BeOS
- MacOS vs. NextOS
- Digital VMS vs. UNIX
- Windows 95 vs. Linux (with XWindow)

Other operating systems that you might consider would include Mach, OS/2 Warp, etc.

Deliverables: The project deliverable is a concise, focused, well-written paper on your topic, limited to **a maximum of four single-spaced typed pages**, plus your bibliography and references.

A Structured Approach to Text Book Selection

H. Roland Weistroffer and Lynda Hodgson
Virginia Commonwealth University

The selection of an appropriate text book for a university level course is an important determinant for the success and quality of that course. Unfortunately, the selection process is often perfunctory, with last minute decisions made based on quickly paging through available text book alternatives. This paper describes an attempt at using a formal procedure for text book selection, based on the analytic hierarchy process and utilizing the software package Expert Choice. The process described includes the identification of relevant selection criteria, the identification and evaluation of candidate text books, and the quest for consensus among multiple decision makers in cases where sections of a course are taught by different instructors. The course used in this case study is a first course in systems analysis and design for third year undergraduate information systems students and involves three decision makers. However, many aspects of the model described apply to other courses in other settings. The hierarchical model was designed in such a way that the course specific part is separable from a more generally applicable part. The experiences in implementing and using the model are discussed.

Introduction

Text books are generally considered to be an integral part of teaching, and thus the selection of an appropriate text book is an important decision process, the outcome of which can strongly affect learning (Farr 1987). Various approaches to choosing a particular book have been identified. These approaches include keeping the currently used text book, choosing a text authored by a colleague, selecting a book based on relevant reviews and comparisons, and voting within the department or among the faculty teaching a certain course ("Lets ask..." 1987). Clearly, these approaches are not all equally desirable.

Since faculty in university business schools (among others) are charged with teaching students how to make good decisions, they should be expected to use good decision techniques themselves, especially when making decisions that affect the learning process. Thus, a rational approach (or at least a structured approach) based on published decision models seems appropriate.

The textbook selection process can be considered to have three major components:

* Identifying relevant selection criteria and prioritizing these criteria.
* Identifying and evaluating candidate books using these criteria.
* Reaching agreement among the involved parties on both of these components.

The objective of this project is to develop a process for the general problem of textbook selection based on an existing decision model, the analytic hierarchy process (AHP), and to validate the feasibility of this approach through a specific application.

The first two decision making components in the textbook selection problem, as listed above, are addressed directly by AHP. Furthermore, AHP also provides assistance for the third component, group agreement. As described by Saaty (1980, 1990), AHP provides a framework that supports multi-criteria decisions. It has been used in a variety of different applications, including strategic planning (Arbel and Orgler 1990), facilities planning (Benjamin, et al. 1992, Carlsson and Walden 1995), and vendor/distributor selection (Davies 1994). Some of the applications using AHP involve decisions made in a university setting, such as evaluating research papers (Liberatore, et al. 1992) and selecting faculty (Dyer and Forman 1992).

AHP is able to structure complex problems that involve subjective judgments, while maintaining simplicity. The ability to structure a complex problem and then focus attention on individual components amplifies decision-making capability (Liberatore 1988). In comparing five conceptually different approaches for multi-criteria weighting, Schoemaker and Waid (1982) found that AHP was perceived as the easiest method to use, and the one whose results were considered most trustworthy. In a later study, Zapatero, Smith, and Weistroffer (to appear) found that Expert Choice, an AHP-based package, was one of the highest rated among five packages for confidence in the procedure.

The rest of this paper will:

* provide an overview of the AHP methodology,
* describe the actual use of Expert Choice in the decision process,
* describe the resultant model of this process
* and discuss some general conclusions derived from this research.

The Analytic Hierarchy Process

AHP allows decision makers to structure a complex problem as a decision hierarchy, with the ultimate decision goal at the apex. At the next level are the decision criteria that contribute to the goal. AHP allows multiple levels of criteria. Thus, general criteria of the same relative magnitude or scope can be identified at one level, then some of these criteria can be broken down into sub-criteria at the next level. Decomposition into further levels of sub-criteria can continue until all contributing factors have been identified and assigned an appropriate place in the hierarchy. At the bottom level of the hierarchy are the actual decision alternatives.

Once the criteria have been identified, they must be evaluated for their relative contribution to the goal or the next higher-level criterion. These subjective evaluations are turned into quantitative weights so that more important criteria can be given more consideration in making the final decision. The software package Expert Choice, which implements AHP, provides for a variety of user interfaces to turn subjective judgments into weights. The user has the choice of directly entering numerical weights for each criterion, or Expert Choice will derive these weights by requesting judgments in the form of pair wise comparisons between criteria.

In pair wise comparisons, every criterion is evaluated by comparing it to every other criterion that has the same parent node in the hierarchy. The user states, verbally, graphically, or by numerical scale whether a criterion is judged to be very strongly more important, strongly more important, moderately more important or equally important. Expert Choice turns the evaluations into a quantitative scale. Equal importance is rated low (1) on the scale, and high relative importance is rated high (9) on the scale. The result is a matrix as shown below:

	Criterion A	Criterion B	Criterion C
Criterion A	1	2	6
Criterion B	1/2	1	3
Criterion C	1/6	1/3	1

In this example, Criterion A is equally to moderately more important than Criterion B, but is strongly more important than C. Criterion B is moderately more important than C. If the relationship is reversed, reciprocal numbers are used, i.e. since the relative importance of Criterion A to Criterion C is 6, the relative importance of C to A is 1/6. The diagonal of the comparison matrix always consists of 1s, and the bottom left of the matrix is always the reciprocal of the top right.

The relative scores in the table should be consistent with each other. Consistency has two meanings. The first is that similar ideas or objects are grouped according to homogeneity or relevance, and the second is that the intensities of relations among ideas based on a particular criterion justify each other in some logical way (Saaty 1990). Consistent judgments allow the software to derive weights so that the sum of the weights for the contributing criteria (or sub-criteria) sum to 1. However, in realistic situations, subjective judgments can be expected to be somewhat inconsistent. Expert Choice determines weights by calculating eigenvalues of the comparison matrices. The package also provides an inconsistency index. Inconsistency index values of .1 or less are considered acceptable (Liberatore 1988). With Expert Choice, the decision maker can simply revise the comparisons, or can request suggestions for improving consistency. The process of improving consistency helps the decision-maker think through and clarify the reasons for making a particular judgment or set of judgments.

Similarly to evaluating the criteria (or attributes) with respect to their relative impact on the overall objective by pair wise comparisons, each alternative at the bottom of the hierarchy is evaluated according to its relative contribution to each attribute (criterion), also by pair wise comparisons. Figure 1 shows an example of a three level hierarchy (i.e. no sub-criteria).

The evaluations of the alternatives with respect to each attribute and the evaluation of the attributes are then combined to get a utility measure for each alternative, i.e. a measure which indicates the degree to which each alternative satisfies or contributes to the overall objective. Expert Choice allows the user to view the results graphically and numerically.

The pair wise comparison technique works best when the number of alternatives is small, generally seven or less. However, pair wise comparisons become impractical with larger numbers. For example, if there were 50 alternatives to be compared, the decision maker would be required to perform $n(n-1)/2 = 1225$ pair wise comparisons (Liberatore 1987). An alternative to pair wise comparisons is to judge each criterion on its own merit.

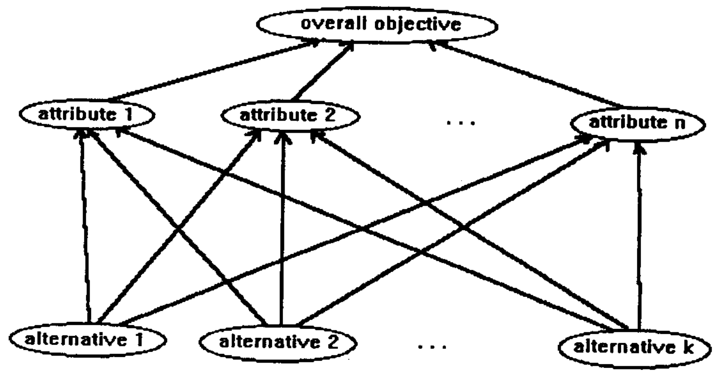


Figure 1: An Example Hierarchy

Group Decision Making with AHP

At the university level, the choice of a textbook often involves multiple faculty members or a selection committee, complicating the decision process. Liberatore, et. al. (1992) note that "academic committees can rarely achieve consensus on anything expeditiously." The problems they encounter include the difficulty of agreeing on criteria, excessively time-consuming discussions, and the challenge of maintaining congeniality among committee members. AHP has been shown to save faculty time when used for the process of evaluating research papers. It has also been successfully used to gain committee consensus on the criteria for allocating funds to improve effectiveness in a university business school. A faculty selection committee used AHP to turn a meandering discussion of potential faculty members into a focused analysis that quickly identified the leading candidate (Dyer and Forman 1992).

Two types of group consensus building using AHP have been proposed. Saaty (1990) suggests that the tool be used in group sessions, where all opinions are recorded and discussed until a consensus judgment is determined. When there is disagreement, this can be resolved by selecting the judgments that are most consistent with the judgments on which there is general agreement. The other approach is to simply average the judgments of all participants (Liberatore, et. al. 1992). The first method requires more time, but hopefully allows all participants to feel comfortable with the final results. The second method probably is more expeditious.

Application of AHP to Textbook Selection

In order to accommodate the busy schedules of all participants, an approach to AHP was used that allowed each decision maker to work as independently as possible. Individual responses were gathered by a graduate assistant who acted as the process administrator. (This person was not one of the decision makers, and had no stake in the final outcome.) Each faculty member (i.e. decision maker) identified decision criteria that he considered important. These lists of criteria from all parties were consolidated into a single list, and shown to all participants for possible modifications (in a way, similar to a Delphi process, but less formal). A likewise procedure was adopted to arrive at a list of candidate text books.

Identifying Relevant Decision Criteria

The decision process began with a short interview session with each of the three faculty members involved in choosing the new text. They were asked to specify both, general text-book selection criteria, and criteria that would be relevant specifically to the choice of a systems analysis and design text. The responses were compared and integrated into a decision hierarchy, with general criteria at the top level (just below the overall objective), and specific criteria at the lower levels. The intention was to clearly separate course specific criteria from those that would be part of a general text book selection model.

Expert Choice was used to structure the criteria. This software package allows for only eight sub-criteria for any criterion or goal, and therefore the criteria had to be organized to accommodate this limitation. This organization process proved beneficial to the overall model because it forced further clarification of the terms used by the three decision makers, and some duplication of concepts was removed. Improvements were also made by grouping related criteria into higher levels of abstraction. Thus the criteria "instructor's manual", "example test questions", and "presentation materials" were grouped to be sub-criteria under the umbrella criterion "instructional support". The result was two levels of general textbook criteria. Criteria that are specific to systems analysis texts were added at the third level. The resulting hierarchy is shown in Figure 2. Note that the same specific criteria needed to be evaluated for both the breadth and the depth general criteria. This was done because the decision makers felt that it was important to cover certain subjects (breadth), but could be less important to include a significant amount of detail on some of them (depth).

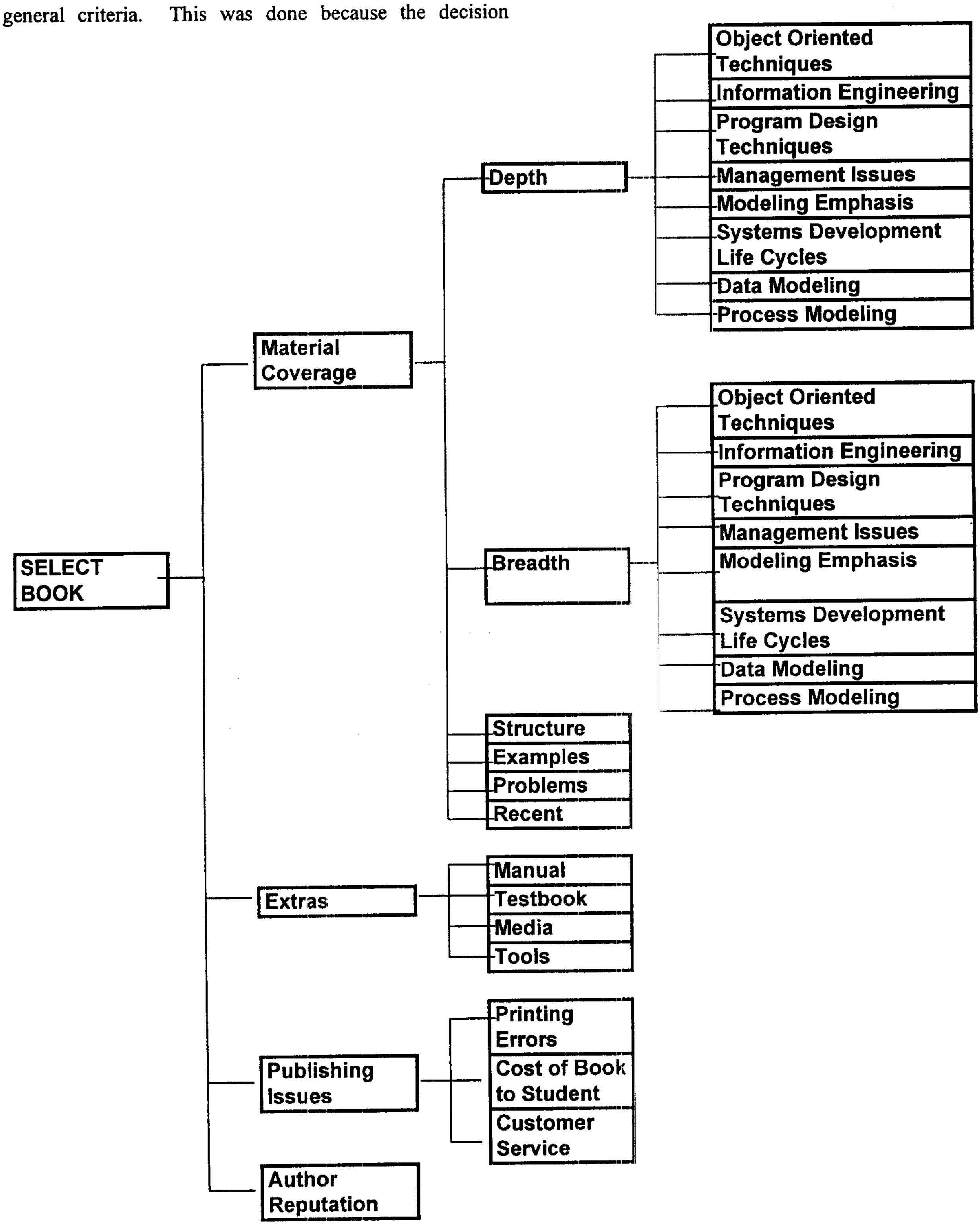


Figure 2: The Text Book Evaluation Criteria Hierarchy

For the weighting process, the model was used to create three independent models -- one for each decision maker. The three decision makers then worked completely independently using their personal model. All of them were allowed to choose the weighting process with which they felt most comfortable. One of the decision makers chose pair wise comparison, while the other two preferred to directly supply numeric weights.

Both weighting techniques resulted in modification of the original set of criteria, because the weighting process provided new insights into the model. It was discovered that certain criteria were actually elimination criteria. Books that did not meet these readily identified criteria were automatically rejected by all of the decision makers. It was agreed by all three that no texts over three years old or texts based on any proprietary vendor tool or methodology would be considered. One of the specific criteria, data flow diagramming, was renamed process modeling to reflect its intent more accurately. All three independent models were updated to reflect these changes. One of the decision makers then decided to restructure the model to contain two levels of specific criteria instead of one. This person also chose to eliminate a few criteria that he felt were not relevant for his personal decision process. These differences were not significant enough to force a reworking of the other two models. Instead, the eliminated criteria were given weights of zero and the other items were rearranged making sure the number and names of all final leaves were consistent across the three models. The ordering and number of levels should not affect the outcome, as long the leaves are given the appropriate weights.

Evaluating Candidate Books

Candidate texts were nominated after the weighting process was completed. Each decision maker was encouraged to nominate two or three texts. Because of the eight sub-level limitation of Expert Choice, as well as the time required to read through and rate each text, it was necessary to keep the number of nominations from each participant relatively small. Six texts, including the one currently in use, were nominated.

It was decided that because the number of final criteria was relatively high, pair wise comparison to rate the texts was not practical. Using pair wise comparison on six books (B) for each of 28 criteria (C) would have required $C*((B*(B - 1)/2) = 420$ independent evaluations. Instead, a rating sheet was developed which allowed the books to be rated on a simple one-to-ten scale for each of the final criteria. This reduced the number of evaluations to $B*C = 168$. Each decision maker was provided with a rating sheet, and copies of the nominated texts. When the sheets were filled out, each one was used to complete the rater's model.

Group Agreement

The initial results showed that there was minimal overall agreement on most of the texts. No books were in the top two across all decision makers. One book appeared in the top 50% for all decision-makers. Other than this there was no commonality in the rankings.

Further inspection of the responses shed some doubts on the reliability of one set of responses. It appeared that the rater simply rated the current text as above average in essentially all categories, and rated all other texts average or below average in all categories. The bias was made clear by two particular responses. In these, the current text was rated above average in the "object oriented" material even though it mentioned object orientation only briefly in the text. The two competing books which were exclusively devoted to object orientation were rated low in this category!

At this point, it was decided that another set of weights and rates should be solicited, so that there would be at least three unbiased viewpoints contributing to the decision. A fourth decision-maker was identified and agreed to participate. Unfortunately, the results were only marginally better. This decision maker made a good-faith effort at providing criteria weights, and provided book ratings for the two books with which he was familiar. However, this decision maker did not rate any of the other books. There were two reasons cited for this. First, he did not feel he had time to adequately review the unfamiliar texts. Second, he felt that the selection process was unnecessary because he had already chosen a new book (one of the candidate books in this evaluation), and was currently using it in his classes.

In spite of these problems, the responses were examined to determine if some conclusions could be derived. Further analysis was done to determine where disagreement existed.

To determine whether there was agreement on the text ratings, the criteria weights were averaged and used to recalculate each decision maker's model. This was accomplished by exporting each of the rater's models into an Excel spreadsheet (a function built into Expert Choice). From this, an average was computed for each criterion. A new Expert Choice model was built by setting up a simple two-level hierarchy that contained all of the criteria with values corresponding to the Excel averages. Then each of the decision maker's text book ratings were applied to the model separately. In this way, if the outcomes were similar, it could be assumed that the major differences were in the criteria weights, rather than the textbook ratings. However, if the outcomes were still significantly different, then it could be assumed that there was little agreement on the textbooks. A similar procedure was performed with average textbook ratings applied to each of the decision maker's set of criteria weights. This was done to determine if major differences existed in the criteria importances instead of, or in addition to the differences in the textbook ratings.

The results seemed to indicate that the disagreement was mainly in the ratings of the books, not in the ratings of the criteria. Using average textbook ratings, but individual criteria ratings resulted in almost identical rankings of the textbooks among all four decision makers. Only places three and four among the six books were reversed for two of the four decision makers.

Conclusions

In this paper we described a structured approach to selecting a text book, making use of the analytic hierarchy process and the software package Expert Choice. As stated in the introduction, the selection process usually consists of three parts, i.e. determining the selection criteria, evaluating the books with respect to the selection criteria, and reaching consensus among multiple decision makers. The first part was accomplished successfully in this project, resulting in the model of Figure 2. This by itself can in general be very helpful in any text book selection process. The second part was only partially successful. Reasons for this may be the busy schedules of the faculty members involved, and perhaps a relatively low priority assigned to this activity by these faculty members. There also seemed to be reluctance (almost fear) by some of the decision makers to commit themselves in such a formal process. The comment was made that to fairly evaluate the text books in this way would require using each of the text books for teaching an actual class first. The third part, reaching consensus, really had not much of a chance to succeed, as it depended on the success of the other two parts. Despite these limitations, the authors feel that the approach is useful, at least in cases of only one decision maker. Success in the case of multiple decision makers would likely depend on the degree of commitment by all members of the group to the application of this approach.

References

Arbel, A. and Orgler, Y.E. (1990). "An application of the AHP to bank strategic planning: the mergers and acquisitions process." *European Journal of Operational Research*, 48(l), pp. 27-37.

Benjamin, C.O., Ehrie, I.C., and Omurtag, Y. (1992). "Planning facilities at the University of Missouri-Rolla," *Interfaces* 22(4), pp. 99-105.

Carlsson, W. and Walden, P. (1995). "AHP in political group decisions: a study in the art. of possibilities," *Interfaces* 25(4), pp. 1429).

Davies, M. (1994). "A multicriteria decisions model application for managing group decisions," *Journal of Operational Research Society*, 45(l) pp. 47-55.

Dyer, R.F. and Forman, E.H. (1992). "Group decision support with the Analytic Hierarchy Process" *Decision Support Systems* 8(2), pp. 99-124.

Farr, R. (1987). "Textbook selection and curriculum change," *The Journal of State Government* 6(2), pp. 86-91.

"Let's ask the students what textbooks they want to use." (1987). *Marketing News* 21(14), p. 24.

Liberatore, M.J. (1987). "An extension of the analytic hierarchy process for industrial R&D project selection and resource allocations", *IEEE Transactions on Engineering Management* 34(l), pp. 12-18.

Liberatore, M.J. (1988). "A Decision support system linking research and development project selection with business strategy", *Project Management Journal* 19(5), pp. 14-21.

Liberatore, M.J., Nydick, R.L. and Sanchez, P.M. (1992) . "The evaluation of research papers (or how to get an academic committee to agree on something)." *Interfaces* 22(2).

Saaty, T.L. (1980) *The Analytic Hierarchy Process,* McGraw-Hill, New York.

Saaty, T.L. (1990) *The Analytic Hierarchy Process - Planning, Priority Setting, Resource Allocation.* RWS Publications, Pittsburgh, Pennsylvania

Schoemaker, P.J.H. and Waid, C.C. (1982). "An experimental comparison of different approaches to determining weights in additive utility models", *Management Science* 28(2), pp. 182-196.

Zapatero, E.G., Smith, C.H. & Weistroffer, H.R. (to appear). "Evaluating multiple attribute decision support systems", *Journal of Multicriteria Decision Analysis*.

Continuous Curriculum Improvement

Through

Alumni Assessment

Earl Chrysler, DBA, CCP
School of Business
Quinnipiac College
275 Mount Carmel Avenue
Hamden, Connecticut 06518

Abstract

This paper discusses the need for outcome assessment for both the purpose of accreditation and assuring the relevance and value of the courses in an Information Systems curriculum. Several methods of evaluating the content of courses are presented and discussed. Alternative questionnaire designs and audiences are presented and discussed. A questionnaire design is proposed for distribution to graduates of Information Systems programs. The questionnaire is designed to allow statistical analysis of the relative value of required and elective courses to a person in an entry-level position and as the person progresses into different types of positions and with the passage of time.

Background

Over the past few years there has been increasing pressure for higher education to demonstrate that it is effective in delivering persons who have the claimed skills and knowledge.

The American Assembly of Collegiate Schools of Business has also incorporated this philosophy into its accreditation procedures. Instead of measuring "inputs" in terms of the qualifications and scholarly performance of faculty, it is requiring Schools of Business to provide evidence that graduates are being provided with the experiences necessary to develop the skills and knowledge promised in the educational objectives that flowed from the School of Business conceptual objective or mission statement. The purpose of this paper is to discuss some methods of evaluating how well the skills and knowledge being acquired by students meet recruiters' needs in the Information Systems field and to propose a technique for the frequent and continuing review of the effectiveness of an IS curriculum.

If one is interested in objective measures that are capable of determining the relative effectiveness of an IS program in general, one can use the following types of measurements:

1. The average and range of starting salaries;
2. The average number of job offers received by students;
3. The percentage of students that have accepted job offers by graduation and
4. The industry standings of the firms that have hired graduates.

This information can be compared to the same information for the Information Systems program for previous years and for the current year for Information Systems programs at comparable regional Schools of Business. To the extent that an IS program is scoring well in the above categories, one might assume that, in general, the Information Systems program educational objectives are somewhat in line with organizations' needs.

However, the most effective method of determining if an Information Systems program is meeting its educational objectives is to examine the quality of its graduates in terms of the skills and knowledge they possess at graduation. There are many ways of accomplishing this.

One method would be to administer some type of super-final exam to all students. Aside from this being an onerous task, several questions could be posed. Does the fact that a student "fails" a test imply something about the program, the student and/or the grading policies extant within the School? If the students pass the exam, what if the skills and knowledge tested are inappropriate for the Information Systems field?

Another method would be to survey firms recruiting graduates of an Information Systems program. The survey form could inquire as to how well graduates are meeting the expectations and/or needs of the employer. This approach also has shortcomings. An employer may not wish to alienate an Information Systems area by being critical of its graduates. Also, the author has observed that on some occasions employers are concerned with relatively short-term goals. They wish to hire graduates who are productive from the first day on the job, i.e., they were trained rather than educated, and are less concerned with the contributions that may be made later in the employee's tenure. There is also the consideration of who is evaluating the person. If the evaluator is the direct supervisor of the person, the evaluation may be much more valid than if the evaluation is performed by a person several organizational levels above the person. Also, if the evaluation of the direct supervisor is edited by one or more persons at a level above the direct supervisor the results may not be valid. In addition, the timing of the evaluation is important. If the person is evaluated shortly after joining an organization, the evaluation will be based more upon the person being able to contribute immediately to the success of the organization. If the person is evaluated some time after joining the organization, the assessment may be based upon other factors. Those factors may be loyalty, interpersonal skills, political astuteness and being a team player, rather than the ability to focus an Information Systems education on specific organizational problems.

Still another method of assessing the effectiveness of an educational program is to have the program evaluated by the graduate. The graduate is the primary "consumer" of the educational experience, having invested the equivalent of four years of one's life and a considerable sum of money in his/her program of choice. It is the graduate, it seems, that knows best if he/she was appropriately prepared for the functional area chosen as a career field. Also, if asked the correct questions, the graduate would most likely respond objectively. One approach is that used by Gasen et all (1) which asked students to evaluate various aspects of the IS program. This study was primarily focused on the satisfaction students perceived with the entire IS program and did not address the content value of individual courses. As a consequence, the results of administering this survey form to graduates would not be very useful in determining whether specific courses should be continued. Also, there were no open-ended questions that would allow the respondents to offer suggestions as to topics not covered that would be valuable contributions to the IS program.

The questionnaire proposed by Hanchey (2) would provide data on several demographic variables, but would require a great deal of time by a respondent and, therefore, the return rate would most likely not be very high. In addition, the graduate is asked to indicate the value of specific IS topics in addition to the quality of preparation in several areas such as analytical skills, values and ethics, etc. Once again, the graduate is not able to evaluate specific course content as to applicability. Also, some information learned in an IS program may have high value in an entry level position, but less later on. However, other information may appear to have little value when a person is in an entry level position, but become quite important in due time.

The Problems

A potential area for a problem may be found when one contacts the alumni department on a campus and determines what information, other than name and address, for a student are on a student record. In most cases the record contains the School or general area from which the student graduated, e.g., Engineering, Behavioral and Social Sciences or Business Administration. However, in some instances this is as much information on the student's course of studies as is available. If one is more fortunate, the area of concentration in Business, e.g., Marketing, Finance or MIS is carried on a student record. If this information is not on a student record, faculty from the functional area will have to attempt to identify from all Business graduates, which ones concentrated in their area.

Another cause for concern is the alumni file updating procedures. When a student graduates, certain information is typically passed from the Admission and Records or Registrar's area to the alumni relations area, including current address. If the alumni relations area is not aggressive about sending out inquiries to graduates as to their current address, and then updating their alumni records with new addresses, many graduates will not be locatable. While one may use the PeopleSearch feature on the World Wide Web or some similar software in an attempt to determine the current address of a graduate, one becomes quickly aware of how many persons share the same name. In most cases, if the alumni relations area does not have a graduate's current address, that person will not be accessible for research.

The next, and critical, issue is what questions to ask and how to ask them, in order to gather the information one needs to determine the effectiveness of a curriculum.

Proposed Solutions

A primary concern is the questions to be asked. Directly related to that is what type of scoring technique to use. The questions must be phrased such that the course being evaluated by the person is viewed without regard to the differing presentation methods of faculty or, indeed, the personality of the faculty member offering the course. This issue should be addressed in the material that introduces the person to a series of courses to be evaluated. By repeatedly focusing the person's attention on the course content, a psychological set would be in place. This should result in the person performing the evaluation by reviewing the course from the frame of reference of the extent to which the course content has been of value in the person's performance.

The scoring issue is always of importance in this type of research. The use of a continuum with numeric values placed at equal intervals on the continuum, we all realize, does not insure ratio scale, or even equal interval, data. Nonetheless, this type of scaling has been used in much research and it appears to be the best available method in this type of situation.

In using this type of scoring, referred to as Likert scaling to identify its originator, if one uses an odd number of values it has been found that there is tendency for the values selected to cluster about the middle value. The implication, therefore, is that by using an even number of values a person selecting a score is forced to lean toward one direction or the other direction, which may result in subtle differences being identified.

The words used at each end of the continuum, once more, are selected so they reinforce the fact that what is of importance is the value of the course content. An example of how a course to be evaluated would appear on a questionnaire is shown below.

MIS 135 COBOL Language Programming

Extremely Weak Content Value	1---2---3---4---5---6	Extremely Strong Content Value

A significant problem is not only the questions to be asked, but the framing of the questions regarding the value of the components of a curriculum at a specific time in one's employ and the type of position in which one finds oneself.

It is desirable not only to obtain an objective assessment of the topics of which each course consists, but to link the value of the course topics to a specific environment in which the applicability of the course topics are being evaluated. As an example, if a person graduated with a major or specialization in the field of Marketing and is currently employed as a sales representative, the person would most likely not ascribe a high value to the content of a course in Marketing Research, whereas a person working in the Marketing Research area of a firm would probably assign a low value to a course in Sales Management.

Similarly, certain course-work that emphasizes the tactical or strategic implications of management decisions would more than likely not be perceived as having a high value to one who had only been employed in a field for one or two years, but would be assigned a significantly higher value if the person had been employed for four or five years and had risen to a management position.

Examples of the use of an introductory paragraph that sets the stage for the evaluation of courses at different stages of one's employment are presented below.

Value of Course Content Upon Graduation

> For each course, circle the number that indicates the value of the course content to you during your first year on the job.

Value of Course Content In One's Current Position

> For each course, circle the number that indicates the value of the course content to you in your current position.

By noting the length of time from graduation and/or the difference in the type of position held immediately upon graduation versus the time the person is completing the survey, the value of the course content can be related to both the passage of time in a given environment and the changing performance needs as a person takes on positions of various types. Sample pages of such a questionnaire will be found as an appendix to this paper.

In terms of the analyses possible, there are several that the reader will quickly perceive upon reviewing the sample pages in the appendix. Some examples are as follows:

Analysis of variance of scores in the Required Courses group to determine whether all courses are evaluated as being of approximately the same value. The confidence intervals for the means of the individual course scores will allow the researcher to identify those courses that appear to have the highest and lowest perceived value.

Another analysis of variance performed on the scores of the elective courses will provide similar insight into the relative value of specialty courses selected by the students beyond the group of courses required of all IS majors.

To determine the change of perceived value of a course over time, one would compare the mean score of each course during the first year on the job versus the perceived value during one's current position after a specified period of time had passed. This would be done for the required as well as the elective courses.

Conclusions and Recommendations

The need for assessment of the relevance and relative value of each course in an Information Systems program is required to develop evidence that is valuable for marketing one's major to prospective students. Perhaps of more importance is the feedback to the faculty within the IS group. The faculty receive information that their curriculum is or is not on track. If not precisely on track, they learn where fine tuning of the curriculum or the content of a specific course is indicated.

The concept of continuous quality improvement is even more applicable. It is the responsibility of a dedicated faculty to continuously be evaluating the curriculum of its area and taking corrective action implied by the assessments supplied by the graduates The procedures presented above demonstrate that this is eminently practical and are shared with readers of this paper to assist them in performing such external assessments on a recurring basis. The author is currently analyzing responses to questionnaires with the described format from Information Systems students with from one to ten years of industry experience after completing the MIS program at California State University at Chico, California.

References

1. Gasen, Jean B., H. Roland Weistroffer and Kimarie Haynes. "Development of an Instrument for Assessing MIS Majors." Journal of Computer Information Systems, Winter, 1991-92, pp. 20-22.
2. Hanchey, Cindy Meyer. "An Assessment Model for Surveying Graduates." Journal of Computer Information Systems , Winter, 1995-96, pp. 67-75.

Appendix

MIS OPTION QUESTIONNAIRE

Value of courses in your first year on the job

Please evaluate each of the following seven required courses as to its content. For each required course, circle the number that indicates the value of the course's content to you during your first year on the job. If you substituted a course for a required course, please print the course title, as you remember it, on a line provided and indicate the value of its content in the area provided. If you did not take a listed course, please indicate that in the appropriate area and do not evaluate the course.

CSCI 052 COBOL Language Programming

Extremely Weak Content Value 1------2------3------4------5------6 Extremely Strong Content Value ____ Did not take

MINS 111 Accounting Information Systems

Extremely Weak Content Value 1------2------3------4------5------6 Extremely Strong Content Value ____ Did not take

MINS 112 Structured Systems Analysis

Extremely Weak Content Value 1------2------3------4------5------6 Extremely Strong Content Value ____ Did not take

MINS 116 Software Project Management

Extremely Weak Content Value 1------2------3------4------5------6 Extremely Strong Content Value ____ Did not take

MINS 118 Data Base Concepts

Extremely Weak Content Value 1------2------3------4------5------6 Extremely Strong Content Value ____ Did not take

MINS 210 Structured Systems Design

Extremely Weak Content Value 1------2------3------4------5------6 Extremely Strong Content Value ____ Did not take

MINS 216 Data Center Administration ______

Extremely Weak Content Value 1------2------3------4------5------6 Extremely Strong Content Value ____ Did not take

____
Substitute Course Title

Extremely Weak Content Value 1------2------3------4------5------6 Extremely Strong Content Value

____
Substitute Course Title

Extremely Weak Content Value 1------2------3------4------5------6 Extremely Strong Content Value

Value of courses in your current position

For each required course circle the number that indicates the value of the course's content to you in your current position. If you substituted a course for a required course, please print the course title, as you remember it, on a line provided and indicate the value of its content in the area provided. If you did not take a listed course, please indicate that in the appropriate area and do not evaluate the course.

CSCI 052 COBOL Language Programming

Extremely Weak Content Value 1------2------3------4------5------6 Extremely Strong Content Value ____ Did not take

MINS 111 Accounting Information Systems

Extremely Weak Content Value 1------2------3------4------5------6 Extremely Strong Content Value ____ Did not take

MINS 112 Structured Systems Analysis

Extremely
Weak 1------2------3------4------5------6 Strong
Content Value

Extremely
Strong
Content Value

____ Did not take

MINS 116Software Project Management

Extremely
Weak 1------2------3------4------5------6
Content Value

Extremely
Strong
Content Value

____ Did not take

MINS 118Data Base Concepts

Extremely
Weak 1------2------3------4------5------6
Content Value

Extremely
Strong
Content Value

____ Did not take

MINS 210Structured Systems Design

Extremely
Weak 1------2------3------4------5------6
Content Value

Extremely
Strong
Content Value

____ Did not take

MINS 216Data Center Administration

Extremely
Weak 1------2------3------4------5------6
Content Value

Extremely
Strong
Content Value

____ Did not take

____
Substitute Course Title

Extremely
Weak 1------2------3------4------5------6
Content Value

Extremely
Strong
Content Value

____
Substitute Course Title

Extremely
Weak 1------2------3------4------5------6
Content Value

Extremely
Strong
Content Value

When you graduated, what was the line of business of the organization in which you had your first job? For example, consulting, system integrator, computer chip manufacturer, computer equipment manufacturer, auto manufacturer, county government

____

What was your job title?

____

In your first job, did you work in the MIS area?

Yes ____ No ____ Partially ____

When did you graduate? Semester __________ Year _______

What is the line of business of the organization in which you work now?

____

What is your job title now?

____

Are you currently working in the MIS area?

Yes ____ No ____ Partially ____

Copyright - Earl Chrysler, 1996

The Oracle Saga: Stories from the Bleeding Edge

Patricia B. Lash, Ph.D.
PLASH@BU4090.Barry.edu
(305) 899-3523

Richard Burroughs
RBURROUG@BU4090.Barry.edu

James Henson, Ph.D.
HENSON@BU4090.Barry.edu

Barry University
Andreas School of Business
11300 Northeast Second Avenue
Miami Shores, Florida 33161-6695
(305) 899-3500
(305) 892-6412 (FAX)

Abstract

This paper describes one university's experience in implementing the relational database management system Oracle and related client/server development tools to support curriculum revisions. The MIS faculty, with the support of the Academic Computing staff, were able to implement the software in three months. However they encountered several obstacles which delayed the project and threatened its success. Ultimately the software was successfully installed. The story provides lessons for other universities attempting to provide leading-edge technologies in the classroom.

Introduction

Barry University is a comprehensive coeducational institution located in Miami Shores with 1700 full-time day students and 7000 full-time equivalent students. The Andreas School of Business at Barry University consists of twenty-two full-time faculty members and offers undergraduate degrees in six areas of specialization, including a concentration in Management Information Systems. The MIS faculty consists of three full-time faculty members.

MIS courses are taught in three programs through the Andreas School of Business. The MBA program offers a concentration in MIS with five courses (15 hours) in MIS courses. The undergraduate day program is primarily composed of full-time students. Their MIS course work consists of the Introduction to MIS course (which all business students are required to take) and eight courses (24 hours) in their major. The undergraduate evening program, with the same requirements as the day program, is designed for students with full-time jobs who are pursuing their degree. While both the day program and MBA program courses run on a traditional three-semester (including summer) academic year, the undergraduate evening program offers courses during four 10-week semesters.

The Andreas School of Business has no internal computing facilities or laboratories. Instead it relies on the services provided by the Academic Computing Center, which supports all academic and administrative offices on and off campus. The Academic Computing Center has a Digital Alpha 2100 OpenVMS server used for general academic computing purposes, including electronic mail, SPSS and Minitab, as well as providing support for program compilers such as C and COBOL.

Most MIS classes are small, with fewer than 20 students, and are housed in one of six networked classrooms which contain twenty-five computers and a networked printer. The instructor's workstation is equipped with an overhead computer projection system. In addition, students have access to an open laboratory, located in the same building, which provides access to over 50 applications packages and 5 operating systems. The open lab contains sixty networked microcomputers and a number of minicomputer terminals.

In September 1996, the MIS faculty began a year-long review and update of their programs. Over the years the MIS curriculum had drifted towards a computer science orientation. In the undergraduate program, of the eight courses taken in the major, four were programming classes, including PASCAL, ADA, COBOL and Visual Basic. The Database Management and Design course used Microsoft Access and there was no CASE tool in place for the Structured Systems Analysis and Design course. Typically professors teaching this course purchased software bundled with the textbook.

During the academic year 1996-1997 the MIS faculty determined to revamp the undergraduate curriculum using DPMA (now The Association of Information Technology Professionals) guidelines to give the program a true MIS orientation and to better prepare the students for the challenges facing the workforce during the twenty-first century. Courses were redesigned with particular emphasis on providing students a better understanding of sophisticated multi-user, client/server application development tools that they will be using in the working world.

As the new curriculum took shape, it became obvious to the MIS faculty that new software would be required to complement the new curriculum. This new software has, in fact, already been implemented. The 1996-1997 academic year has seen the implementation of Oracle 7.3, availability to the students of UNIX, OpenVMS, Windows NT, Microsoft Project for project management and application development using Visual C ++. This paper focuses on the implementation of Oracle and related applications. It describes how a small school in a small university was able, in the space of three months, to provide its students with state-of-the-art tools as a critical component of their MIS education.

Making the Leap

Most of the MIS graduates from the Andreas School of Business are employed in small to medium-sized businesses with an emphasis on client/server use and client/server development. They generally find themselves in such environments as Novell networks, local area networks, Windows NT servers, Windows NT networks, and UNIX systems and servers. They are increasingly utilizing application development tools based on relational database technology.

As the MIS faculty began to explore the various software options available, they investigated the leaders in that area: Oracle, Sybase, Microsoft's SQL Server and Ingres. A *Computerworld* forecast for jobs in 1997 (Menagh 1997) confirms the importance of Oracle and Microsoft SQL Server in information systems hiring. In reviewing the capabilities of these products the faculty felt that there were a number of products that would serve their needs for their academic efforts at the university. Oracle was particularly attractive primarily because of their Academic Alliance Program which provides the university with access to any of their software on any platform with installation on as many machines as desired for academic and instructional use. The total cost for software and services is a reasonable $500 per year. Oracle also provides a set of manuals and tutorials in addition to their Bronze Level hotline support, which is basically 8-5, five day a week support. Microsoft does have a scholarship program with support for academic institutions, but the faculty was wary of expending the time and effort to go through that process with the prospect of being turned down. They also felt that SQL Server by Microsoft does not share Oracle's widespread recognition as the industry standard product. Given the level of support that Oracle provided at such a reasonable cost, the MIS faculty selected Oracle. As a small program, with only three full-time faculty members and no computer resources completely at their disposal, the faculty were aware of the costs of switching over to the new technologies.

The new MIS curriculum was approved by the Academic Affairs Council in November and scheduled to be implemented in Fall 1997, with the publication of the new catalog. In reality, it would have been possible to delay the implementation of any new software until that time. As faculty began to finalize plans for Spring 1997, however, they began to consider acquiring some software even sooner. One undergraduate section of Systems Analysis and Design was scheduled for Spring 1997 and it was necessary to select some CASE tool to be used in this course. Of greater concern, however, were the graduate courses. A Database Design and Management course as well as a Current Topics in Software Development course were scheduled for Spring. Several of the students in these courses had undergraduate MIS degrees and had already taken one course using Access. Since many students in these two classes were scheduled to graduate before Fall semester, it was decided that they would benefit greatly from some exposure to the Oracle tools. Based on the communication with Oracle and Academic Computing, the faculty determined that it would be possible to be ready to go live with the new software by February.

Implementation

Once the faculty selected Oracle, it was necessary to secure the funds from the Dean, since this was not a budgeted item and had not been planned for in the current academic year. Because the total cost to the Andreas School of Business was only $500, the Dean was amenable. The faculty then coordinated with Academic Computing to put the paperwork in place.

The Academic Computing area is responsible for the OpenVMS server with one systems programmer. Faculty members at the Andreas School of Business do not have systems software management access to that machine. The support and cooperation of Academic Computing as well as their services for the implementation and support of the product once it arrived would be critical to the success of the project.

The Academic Computing Center staff was favorably disposed to provide the MIS area with use of their machines. Several years earlier there had been a somewhat unsuccessful attempt to install Oracle. The Academic Computing staff was eager to have a second opportunity to implement this software. They did not, however, want to be responsible for it. Due to its size and complexity they felt that they were unable to guarantee success with it. One concern was that the MIS faculty have some background in Oracle before they proceeded. One of the faculty members who had managed an MIS organization that had used Oracle offered managerial experience with the software, but not from a technical implementation and support perspective. However, through continuing discussions with the MIS faculty, the Academic Computing staff gained confidence in the success of the project. Once the paperwork was put in place, boxes upon boxes began to arrive from the Oracle Corporation.

To successfully install Oracle once they received the boxes, the MIS faculty discovered implementation was not quite as straightforward as expected. One of the things absent in the promotional literature and in conversations with the Oracle "people" was that the software required 96 megabytes of RAM on an OpenVMS server. The current capacity was 64 megabytes, so an upgrade was required prior to the Oracle installation. Since requirements differ for each machine, it is difficult to draw any conclusions from the omission except *caveat emptor*. However, the need to upgrade the RAM proved to be a real financial setback. The $500 investment suddenly became a $4500 investment because the memory upgrade cost $4000 and, again, this was not a budgeted item and occurred in the middle of the academic year. The upgrade would be installed on machines not under the control of the School of Business and would, of course, be available to other applications on the server. Initially Academic Computing was very supportive and offered to pay for the upgrades; they later backed away and pointed out that this was not a budgeted item and they could not provide it during this academic year. When it became obvious that neither Academic Computing nor the other academic units of the university would financially support the upgrade, the Dean of the Andreas School of Business provided the funds from internal sources.

There were a number of consequential tasks required of Academic Computing before they could really support the implementation of Oracle. The additional memory for the server had to be installed, the operating system had to be upgraded from Version 6 to Version 7 of OpenVMS and a number of other products optioned for Version 6, including Minitab and SPSS, had to be upgraded.

These preliminary preparations lasted from November into the latter part of January. The MIS faculty had promoted the acquisition of Oracle to the student body, and the students were very eager to upgrade their capabilities from Microsoft Access 2.0 for Windows to Oracle 7.3 enterprise relational server technology. However, when classes for the spring semester commenced in January, the code had not even been loaded on the machine yet.

The aforementioned three classes were originally set up to take advantage of Oracle in the spring semester. The MIS faculty designed their syllabi and prepared their course materials so that they would be ready to start using Oracle around the end of February.

Resources for implementation were limited, in the Andreas School of Business as well as in Academic Computing. A senior undergraduate MIS major who had a part-time job at an Oracle value-added reseller and had installed Oracle in several different environments was identified. This student was assigned an independent study with the university to install Oracle. This arrangement proved to be successful to a limited extent. The student was able to install the base Oracle system software on the server within about four weeks, which is not unreasonable for the size and complexity of this software. Clearly, however, academicians are not accustomed to installing and supporting this level of software with this complexity and this time delay and there continued to be many obstacles to overcome.

Once the base system was functional, it was necessary to upgrade the workstations in the computer lab. These were 486/66 PCs running Windows 95 with 16 megabytes of RAM. They had to be upgraded to 32 megabytes of RAM before the CASE tools for Oracle could be supported. Textspace mode was running so that students could sign on via a VT100 interface and add tables, delete tables, etc. using SQL against the server. This was adequate for the Database Design and Management class. The CASE tools, however, were important for the Systems Analysis and Design class, so the students could draw data flow diagrams, entity-relationship diagrams, and functional hierarchy diagrams as part of that course. So the memory was installed, and the software was loaded.

Once the software was loaded, the biggest problem encountered was getting the client software on the Windows 95 PCs to talk to the server software. The process uses a facility called ODBC which links up through an Oracle proprietary product called SQL*Net on the client and SQL*Net Listener on the server running over a TCP/IP network. All of these pieces had to operate correctly for the client/server portion of the technology to work.

After installing everything according to the manuals, the student assistant continued to work on establishing the link between clients and servers. Over a period of six weeks, they moved from having lots of errors whenever they tried to initiate a communication to no errors, but still the software did not work. Apparently something undocumented was interfering with the process. At this point the student intern gave up and said he had absolutely no idea what was wrong, could not figure it out and did not know what to do next. The MIS faculty member who had taken the lead role in the project called Oracle Support, which through the academic program was available Monday through Friday. Oracle personnel immediately began work on the problem. One person worked full-time on the East Coast and another full-time on the West Coast, each dialing into the server and tweaking various options. It took two of them, over four days of solid effort, to figure out the problem, now known in the Andreas School of Business as the *Barry Anomaly*. A number of modifications to the server were required.

Once client/server communication was initiated, Designer 2000, an upper-case tool used for client/server development, was installed. It is a very sophisticated toolset, with some pieces of the application developed for the server and others developed for the client. Prior to Designer 2000 becoming operational, the faculty had had CBTs running on it off-line, but now they were eager to begin using it. There were a number of problems installing the software, but these were overcome in a matter of a day or two. At this point, the MIS faculty were ready to allow the students to access the software.

Near the end of March, one of the graduate classes was able to sign on to develop client/server applications. Almost immediately the software failed. It failed because they had loaded a demonstration repository under Oracle and asked the students to go through a tutorial provided with the product. However, as is to be expected with a client/server database, a student was consistently denied access when another student was updating the database. The problem was solved by installing ten copies of the tutorial, one for each of the ten students in the class.

At the end of March, the MIS faculty were fully operational in a GUI client/server mode. The system could accommodate 25 log-ons with each student able to create, delete, and develop in his or her own workspace with his or her own tables. Oracle has notified the faculty of one outstanding problem using Designer 2000 with Version 7.3.2.2.0 of the server software and are sending a patch to take care of that prospective problem.

Evaluation

One of the challenges with client/server technology is getting everything to work well in a balanced manner. In the last five months Barry University has received a real taste of the complexities and frustrations inherent in such a project. While the students in the spring semester have not been able to use Oracle as much as the faculty would have liked, all involved credit this project as a success. The software is practically completely implemented. The faculty have been very open in terms of the problems that they encountered, so that the students have learned a great deal about the real-life issues of implementing large, complex enterprise-wide client/server technologies. They now recognize the complexities of a real MIS environment and have been trained in the type of environment which will put them a step ahead of organizations that merely train database users on Microsoft Access and Visual Basic.

Academic Computing dedicated the "best machines in the house" as clients. However these workstations were only equipped with 800 megabyte hard drives and about 400 megabytes are required for Designer 2000. Once the faculty have more experience with Designer 2000, they will be able scale back the modules that are installed. However, until the determination of what is absolutely essential is made, there have been some problems with the capacity of these machines. Because these workstations also support MINITAB , SPSS, and C++ as well as Microsoft Office, once all of the software was installed, they only had about 3 megabytes of hard drive available, which creates some problems under Windows 95. One MIS professor donated a hard drive to be installed on the instructor's workstation and on other workstations Microsoft Project was removed in order to free up some space. Because the graduate MIS classes this semester have fewer than fifteen students each, the provision of different complements of software at different workstations has not yet caused problems. However, the teaching lab will require larger hard drives at some point and the MIS faculty is aware of the need to manage disk space.

The impact on the spring classes has been varied, with the graduate classes faring better than the undergraduate. The undergraduate Systems Analysis & Design evening class was only ten weeks long and ended in late March. These students were disappointed that they were unable to use Oracle. Although, it was never really promised to them in advance, once classes began the professor made them aware of the project and was able to use examples from it to illustrate principles discussed in class. These students expressed an interest in taking future courses using the technology, whether as undergraduate or graduate students. For this class, the faculty were relying on having Designer 2000 running and being able to use its tutorial to teach the students how to go through that process. This software was not operational before the ten week semester ended, so this course was not able to use a CASE tool at all
The graduate database class is primarily concerned with application design issues associated with database, which is really product independent. They were able to use the SQL*Plus feature of Oracle prior to the installation of the GUI. Once Oracle was fully installed they were able to use the more advanced tools. In the Current Topics in Software Development graduate course one class was canceled because of the late implementation of Oracle. That class has been one class "late" because of the difficulty in getting the software up and running. However, the students have discussed the problems and issues associated with implementing large client/server development tools . Additionally each student conducted on-line research using the Internet and sent e-mail on ideas garnered from discussion groups, Oracle pages, and other sources available on the Internet.

All MIS professors preach the importance of training in the implementation process. However, in this instance, although the software is operating, the faculty are not really trained on how to effectively use it in the classroom setting. This is expected to be an evolutionary process. By next fall the full-time faculty members will be fully trained on the software. How the adjunct faculty will be trained has not yet been determined.

Lessons Learned

Lesson 1: Cooperation is essential.

The move to the new technology was part of a larger effort to revamp the curriculum, and was done in the spirit of redesigning the entire undergraduate MIS program. One of the primary lessons is the importance, especially in a smaller university with fewer resources, of working with other university units. The cooperation and support between the MIS faculty and the Dean of the School of Business and between the MIS faculty and the Academic Computing group were critical to the success of this project. The support of the Dean, especially when it was necessary to upgrade the server, was critical. The project would have died had he not stepped forward with the requisite funds. Clearly, the Dean went above and beyond what might have been expected. In a small school with somewhat limited resources, it required real courage on the part of the Dean to continue to support these efforts.

While Academic Computing was not as supportive fiscally, they have been supportive in terms of the facilities, time and attention that their staff has paid to the Oracle project. They have given very strong support in terms of the system software, and even the enduser support people have been of assistance in installing the client software for Oracle.

Clearly, the assistance provided by Oracle was critical. When the project group had exhausted all of the expertise available to them locally, Oracle stepped in and made their product work. Given the real need to have the software operational before the end of the semester, Oracle's efforts to solve the problem expedited the process. Internet access was a crucial component of the relationship between Barry University and Oracle.

Lesson 2: Plan carefully.

Implementation of software in an academic setting is not equivalent to business implementations. In this case the MIS faculty were dealing with virtually no resources and were relying on vendor's special programs for academic institutions. Additionally, in an educational setting the timeline issue is a bit more complicated, primarily because educators are constricted by the academic calendar. In this case, although the MIS faculty knew that they were pushing the timeline by attempting a Spring semester implementation, they felt that the benefits to the students of using even a partially installed Oracle product would outweigh the inconvenience.

As Rob and Coronel (1997) point out, client/server development is significantly different from the traditional information system development in both process and style and must be managed differently. Therefore, many of the traditional guidelines may not apply in an academic setting.

However, one model of the development of client/server systems (Rob and Coronel 1997) points out that the first step should be an infrastructure self-study, including software and hardware, critical applications, human resources, and problems and opportunities. In this case, the infrastructure was not adequately considered. Oracle was selected without adequate understanding of the hardware requirements of the software. Additionally the impact of the software on both the capacity of the client and the server was not understood. The real lack of personnel to handle the project hindered the progress. Had the MIS faculty had a more complete understanding of the potential problems, they might have delayed the implementation and taken a more measured approach to the process. While these problems were eventually overcome with much anguish and effort, better planning up front would have led to a smoother implementation.

Lesson Three: Educational institutions require special consideration.

Normal organizations pay Oracle or their VAR to install the software. The also generally require at least $6000 worth of Oracle education. While Barry University saved this $20,000 to $30,000 expense, it did make life more interesting. Many of the obstacles might have been more easily overcome or avoided completely had there been more resources available to installation and training.

Conclusions

The MIS faculty at Barry University can offer the following conclusions based on their experiences:

1. Projects such as this are important to do for your students and the reputation of your program.
2. Even a small school with limited resources CAN undertake such an ambitious project.
3. It is critical to talk to the vendor before committing to the software to be sure that ALL of the implementation issues are identified and addressed.
4. Try to secure community resources (i.e. free installation from a local VAR) to assist in the implementation.
5. Give yourself some time to get the kinks worked out and to learn the product.
6. Develop and share curricula.
7. Remember, it is projects like this that make MIS fun!!

References

Menagh, Melanie. "Experience Required," *Computerworld*, December 23, 1996/January 2, 1997, pp. 79-81.

Rob, P. and Coronel, C., *Database Systems: Design, Implementation, and Management*, Course Technology, 1997.

Professional Issues in Information Systems Education: Background, Critique and Directions for the Future

David R. Arnott and Martin D. Atchison
Department of Information Systems,
Monash University, Melbourne, Australia

Abstract

This paper discusses the importance of the linkages between professional practice in information systems and the work of a university. It critically reviews some of the common forms of professional engagement used by universities in the context of recent critiques of the professions and their place in society. The paper identifies some activities of the major professions which may improve information systems education.

Introduction

This paper addresses an important topic in information systems education - the relationship of universities with industry and the profession (broadly defined as 'professional engagement'). In a professional discipline like information systems, where the vast majority of graduates progress to a career in business and government, these links are particularly important.

The paper is organised as follows: first, it reviews the literature on professional education and identifies information systems as a professional discipline. This is followed by a critical review of some of the common forms of professional engagement used by universities, and comments on the key issues in teaching professional practice.

The Problem of Professional Education

Hughes (1959) described the role of the professions in society in terms of a bargain in which professional practitioners supply their specialised knowledge and expertise to solve societal problems in return for privileges such as status, professional autonomy, and the right to control their profession in terms of conditions of entry, standards of practice and so on. In long-standing traditional professions (eg law and medicine) this bargain is well-established and has been institutionalized in a supporting organizational and legal framework.

Schon (1987) argues that in recent times this arrangement has begun to break down as the professions have found it increasingly difficult to maintain their side of the bargain. Practitioners in almost all professions have found that their ability to solve social problems is being called into question. These doubts have created two crises in professional practice- one concerning the nature of the problems which professionals are called upon to solve, and the other concerning the nature of the special knowledge which they claim to possess.

Schon (1987) suggests that professional practice has a 'high ground' in which practitioners can apply their professional knowledge and skills to solve manageable problems, but it also has a 'swampy lowland' in which 'messy, confusing problems defy technical solution' (Schon, 1987, p3). Practitioners who work in this lowland are confronted with:

- problems for which the task of problem definition is itself problematic, where the problem statement will vary according to the perspective of the person describing it;
- problems which present unique aspects for which no precedent exists in theory and for which no 'text book' solution exists; and
- problems which involve direct conflicts of values between those people affected by it, such that the choice of a solution requires a choice between these values.

Professional practitioners must decide whether to retreat to the high ground where they are confined to a limited set of problems which are of relatively minor importance but for which their technical skills still work, or whether to descend to the swamp where the major problems exist, but where their technical skills are inadequate.

The failure of professionals to address this issue satisfactorily has led to expressions of doubt about the legitimacy of their claim to special status as societal problem-solvers. Critiques have extended from a simple questioning of the professional's ability to serve this function through to more radical attacks which have portrayed professionals as supporters or instruments of the establishment, hiding their political agenda behind a mystique of technical expertise (see for example Illich, 1970 and Larson, 1979).

The second crisis of professional practice described by Schon concerns the nature of the knowledge which practitioners are taught as part of their professional education. Schein (1973, p43) categorised professional knowledge into three components:

1. an underlying discipline or basic science component upon which the practice rests or from which it was developed.
2. an applied science or "engineering" component from which many of the day-to-day diagnostic procedures and problem-solutions are derived.
3. a skills and attitudinal component that concerns the actual performance of services to the client, using the underlying basic and applied knowledge.

Schon argues that the university curriculum has traditionally been weighted strongly towards basic science. Applied science and the technical skills of day-to-day practice have been accorded less prestige and have been less well-catered for in university teaching. In Schon's view, the assumption that academic research in areas of 'basic science' will produce useful knowledge for professional practice is no longer necessarily true. Increasingly, practitioners, educators and researchers alike recognise that the content of university curricula has become less relevant to professional practice (for example, Glazer, 1974; Schein, 1973; and Simon, 1972).

Schon suggests that attempts to fill the gap between the university curriculum's conception of professional knowledge and the requirements of professional practice usually take one of the following forms:

- treating the problem as one of lack of currency of the curriculum, and trying to up-date it to keep up with the latest developments in research;
- treating the problem as one of lack of appropriate content in the curriculum, and trying to include new elements to it which include aspects of professional practice not normally covered in traditional professional education;
- treating the problem as one of lack of rigour in the profession, and trying to tighten professional standards.

In Schon's view, these strategies are stop-gap measures which do nothing to address the real issue:

> Can the prevailing concepts of professional education ever yield a curriculum adequate to the complex, unstable, uncertain and conflictual worlds of practice?
> (Schon, 1987, pl2)

The Discipline of Information Systems

Information systems is concerned with the development and effective use of computer-based information systems and their impact on people and organizations. Human and organizational factors are at least as important as technical factors, and it is this focus which most distinguishes information systems from software engineering and computer science. Information systems is a "professional" or "applied" discipline. Its relatively recent origins make its status as a profession within a strict definition of the term open to debate. According to Moore (1970, p56) a profession '... involves the application of general principles to specific problems', and further (pl41):

> ... the two primary bases for specialization within a profession are (1) the substantive field of knowledge that the specialist professes to command, and (2) the technique of production or application of knowledge over which the specialist claims mastery.

These criteria serve to distinguish a profession from an avocation which is '...based upon customary activities and modified by the trial and error of individual practice' (Moore, 1970, p56). Hence, a discipline's claim to the status of a profession rests in part on its ability to define a body of rigorous scientifically-based knowledge, and in part on its ability to describe a set of techniques by which that knowledge is applied to solve problems.

Although we believe that information systems does warrant the use of the term profession, it is not the purpose of this paper to argue the merits or otherwise of its claims in this regard. However it is important to stress the point that the main purpose of the study of information systems is to improve practice. The vast majority of graduates in information systems progress to a career in practice rather than in academic life. Hence the principal aim of an information systems degree is the education of a professional systems analyst who requires a combination of technical and social skills in order to function effectively.

If it is to thrive as a professional discipline, information systems is strongly dependent on a continuous interchange between scholarship, research and professional practice (Keen, 1987). Within the discipline, information technology is best viewed as an enabling factor on which these three elements rest. This is illustrated in Figure 1 which also highlights the role of reference disciplines in providing relevant theory and research approaches. Practice is shown as being relevant only in an organizational context.

Research is a systematic process of acquiring new knowledge. Research results can feed into scholarship, in the enhancement of theory, and professional practice, through the development of new methods or enhancement of existing methods.

Scholarship is the process of systematizing existing knowledge that is relevant for a discipline. It feeds into research by contributing to the generation of new or revised theories. Scholarship involves logical inference drawn from theories and previous research rather than directly from data. As the development and use of information systems takes place within an organizational context it is important for information systems scholars to be cognisant of the state of its major reference disciplines. These reference disciplines are many and include management, economics, psychology and computer science.

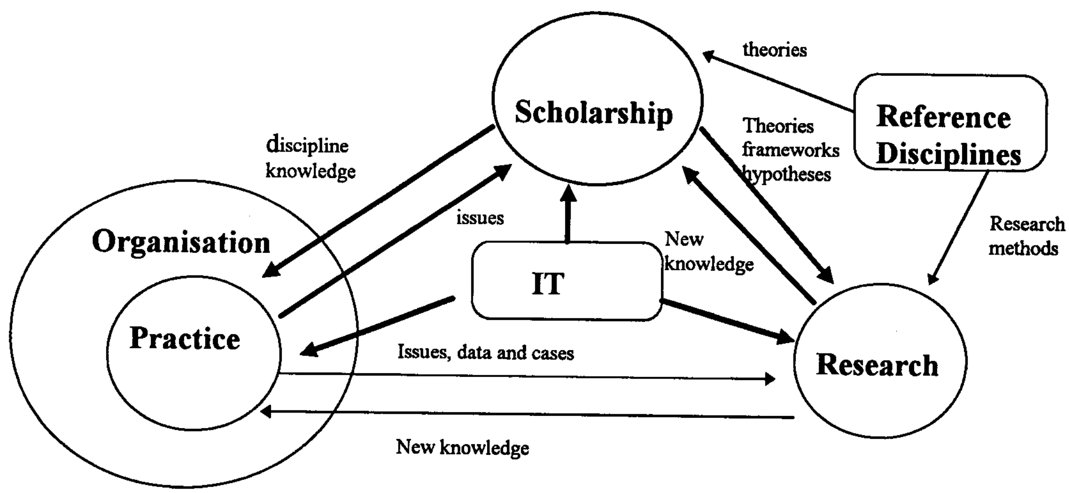


Figure 1. A Model of the Discipline of Information Systems

As depicted in Figure 1, research, scholarship, practice and He growth of knowledge in reference disciplines take place independency from one another. The establishment, development and nurturing of links between These areas is essential to the long-term future of the profession. Our particular concern in this paper is The links between practice and research and practice and scholarship

Professional Engagement by Universities

The links between scholarship and practice, and research and practice can be collectively termed 'professional engagement". In terms of Schein's (1973) classification of professional knowledge quoted in the previous section, they are concerned with the third category of knowledge - the actual performance of services by a professional for a client. These links determine the impact of the university on the profession and the influence of the profession on university activities. The nature of the links can be quite subtle and are almost always bidirectional.

The most common forms of professional engagement fall into two main categories - teaching by edict or direct instruction, in which student learning is based on the delivery of material presented by an instructor, and teaching by doing, in which the students learn by carrying out the activities which they have been taught and reflecting on their experiences in doing so.

(a) Teaching by Edict or Instruction

Professional Accreditation

The most formal link is the professional accreditation process of some designated professional body. Accreditation defines the minimum curriculum requirements for students to become entitled to admission to membership of the body and to use its status and prestige as a justification for claims to special knowledge and expertise. When used appropriately, professional accreditation can act as a valuable means for stimulating debate about the nature of the special knowledge which is central to the discipline, and can provide benchmark standards for curriculum development.

Professional accreditation can also be a conservative influence on information systems curricula. In an industry characterized by rapid change, it is very difficult for professional bodies to maintain anything other than a broad framework for accreditation. An outdated specification of what is required for professional accreditation may constrain the use of research results as the basis for new teaching topics, or may condemn the accrediting body to irrelevance in the eyes of the teaching and practicing community. Universities must be careful to assess the representativeness and importance of professional bodies when faced with a conflict between the edict of a professional body and curriculum philosophy.

Course Advisory Committees

Advisory committees which include prominent practitioners provide another mechanism through which a university can discover what material industry feels is important for a degree program. Such committees also act as vehicles for the university to influence professional practice, by ensuring that leading practitioners are fully informed on the details of recent advances in research and scholarship in the discipline.

Course advisory committees can also be a conservative influence on curriculum, especially when their formal approval is required for new courses and subjects. Practitioners can sometimes take a narrow view of the discipline because their own experience of it is largely confined to what is happening within their own working environment. They may also have an element of self-interest in wanting to establish and maintain courses which support their own organisation's needs and objectives. The overall efficacy of advisory committees depends on the nature of the membership and members' willingness to consider what is required for the foundation of life-long learning and future practice.

Industry Surveys

Universities will often survey industry to confirm current and to elucidate opinion on future trends. These surveys may be quite simple and in-house or they may be rigorous studies of industry trends (Brancheau et al., 1996; Broadbent et al., 1995; Klobas ~ McGill, 1993; Lee et al., 1995; Watson, 1989). Like course advisory committees and professional bodies, they can be extremely valuable as a source of material to guide course and research design.

Depending on the nature of the sample and the instruments used, industry surveys may also be a conservative influence. Long lead times in implementation and the conservative nature of some organizational practices ("avoid leading~dge technologies and methods until they are tried and proven") mean that in some cases industry practice may lag behind advances in the discipline. Using a survey of industry practice as a basis for course or curriculum design requires careful evaluation of its validity as a measure of industry practice and the relevance of its findings to the true state of the art of the discipline. Their quality in these respects is, however, usually better documented and easier to measure than for either of the previous two methods.

Guest Lectures

A simple and popular way of linking subjects to practice is to invite guest lecturers from industry to deliver part or all of a subject. Guest lectures can be extremely effective if the basic theory has aheady been taught and the students have the knowledge to be able to critically assess the practitioner's input.

As with course advisory committees the quality of a guest lecturer's input in this regard is heavily dependent on the personal characteristics of the individual and their motives for involvement. In addition, guest lecturers usually are able to get only a partial view of the full curriculum, and may find it hard to relate their material to the overall objectives and direction of the course.

(b) Teaching by Doing

Realistic Assignment/Project Work

A common way of attempting to assess students' knowledge and skills, while at the same time illuminating the nature of practice, is to make assignment work as realistic as possible by requiring the student to apply the material being assessed in a real-life situation. The extent to which the problem environment described to the student reflects reality can vary from a purely hypothetical scenario created by the teacher through to actual real-life projects where students develop a system or parts of a system for a real client. This gives the student the opportunity not only to learn a professional skill or technique, but also to have some experience of applying it in an environment which bears some resemblance to professional practice.

This approach has obvious appeal but is subject to a number of problems. In order to present the real problem in an assignment, it is usually necessary to simplify the problem specification. In doing so the problem may be trivialized and the students may gain a false impression of the nature of practice, especially in terms of its difficulty. This can lead to significant over-confidence and professional naively in new graduates. At the other extreme (particularly when students do projects for a real client), reality can intrude to the extent that the teaching objectives become obscured; students may be overwhelmed by the complexities of an environment with which they are unfamiliar. Their inability to separate out the elements relevant to them as information systems practitioners may leave them confused and uncertain about where and how to apply their specialized knowledge.

Sandwich Courses

Sandwich courses require students to spend time working in industry - in Australia typically for a year between second and third year in an undergraduate degree. This gives the student the opportunity to work alongside practitioners, and participate in projects where they can apply their learning to real life practice. This is similar in many respects to an apprenticeship system, and clearly has the potential to offer great benefits in exposing students to the realities of professional practice.

The benefits gained during the sandwich year ale obviously heavily dependent on the quality of the placement. In worst cases, students may be exploited as cheap labour for routine office tasks such as photocopying and data entry, which have nothing to do with their future careers as professionals. Assessment of the work done is also difficult particularly given that students may have very little influence over the quality of the work they do and the mentoring they are offered. The presence of a sandwich year also has a subtle effect on course design. Whereas a three year undergraduate degree commonly has an aim of preparing a student for employment, the "sandwich degrees" tend to try to achieve most of this aim by the end of second year. This compression of material may have the effect of reducing the theory content, as the need to have the students performing well in industry colours the design of subjects.

Industry Based Learning

Industry-based learning (IBL) enables students to take placements in industry as a substitute for subjects in a three year undergraduate degree. IBL reduces some of the problems of sandwich courses by providing shorter periods of industry experience spread over a number of years. These programs tend to place more emphasis on assessment than sandwich programs and the links between the university and industry partners are much closer.

Although the severity of the problems of sandwich courses is reduced, the problems are not eliminated entirely. IBL programs still suffer from variability in quality depending on the organizations and practitioners involved. Timing the presentation of theory to complement the practical experience continues to pose difficulties.

Applied Research

The most intimate link between university work and industry is the conduct of applied research, particularly when the object of study is an information system under development. Research with a strong practical focus provides a great opportunity for students to explore in depth the adequacy of theory in the light of the constraints imposed by real-world situations.

Despite the obvious appeal of applied research in principle, the extent of its use in practice has been constrained by the fact that there are many fundamental research questions in information systems which need to be answered before it can be properly exploited. It can be argued that the low impact of information systems research is in part due to the relative lack of fundamental research in the discipline.

The forms of professional engagement discussed above have provided an important foundation for establishing the professional relevance of the discipline of information systems. However, academics need to be aware of the deficiencies of some of these forms of engagement and of the conservative influence that they can exert on curriculum. It is obvious that the quality of these university/profession links are of widely varying quality and they are probably much less effective than universities claim or believe. Many of the links are superficial; they can also be conservative and variable in quality; and some are antithetic to the values of a university.

This state is to be expected given the relatively early stage of development of information systems as a discipline. The development of sound bases for research, scholarship and practice in any new discipline tend to be sufficiently difficult in themselves to distract attention from the need to develop and nurture the links between them. However, if the discipline continues to fail to address these issues, it runs the risk of encountering the kinds of crises of confidence outlined by Schon (1987). There is already evidence to suggest that problems of this sort are emerging in information systems.

Improving Professional Engagement

In terms of developing the links which contribute to better forms of professional engagement, the first step is to improve the existing mechanisms through a process of continuous quality assurance. An important element of this process must be the establishment of a clearer picture of the aims of professional engagement and the ways in which different teaching mechanisms can contribute to their achievement. The implementation of well-intentioned activities of the type described in the previous section will suffer if there is a lack of clarity in their statement of objectives for education in professional engagement, and the way in which the curriculum content meets them. Exposing students to professional practice in an ad hoc way will not contribute significantly to student education in this area.

One of the confusing aspects in any analysis of the links between universities and professional practice is the fact that the use of practical examples to illustrate points of scholarship or research is commonplace. This can give a misleading impression that elements of a university curriculum have a close connection with professional practice, when in fact their contribution to professional engagement is peripheral to their main objective. Existing programs which specifically state their aim to be the teaching of 'professionalism' or 'professional practice' have tended to focus on two areas:

- *specialized professional technical skills:* key technical skills required of information systems professionals which are not specific to the information systems discipline but are general to all forms of professional practice; these commonly include elements such as interpersonal communications, group dynamics, and writing and presentation skills;
- *the professional work environment:* the social and organizational environment within which information systems professionals will practice and the way in which their professional status affects their responsibilities within that environment; these commonly include elements such as professional ethics, the social responsibilities of information

systems professionals, and the role of the information systems professional in organizations.

Although these are certainly a necessary part of the overall requirement for initiating students into professional practice, they are not sufficient, because they deal mainly with the relatively 'mechanical' aspects of practice rather than the behaviours and patterns of thought and action which are characteristic of highly competent practitioners. Schon (1987 and 1991) has written extensively on the need for students to be exposed to models of such behaviour, so that they can learn 'how to be a professional' rather than just learning the skills and techniques which a professional uses. He describes the teaching process requited for this as being more akin to coaching than traditional teaching, and suggests that there is a need for a radical re-thinking of the mechanisms for teaching professional practice. In some older professions such as medicine and law, mechanisms for linking teaching with professional practice already include some elements of this approach.

One approach is to appoint many more senior staff from industry to university departments in an honorary or adjunct capacity. These practitioners should have to satisfy the same criteria for selection as academic staff. There are a growing number of practitioners who are conducting research and publishing the results, and who are teaching at university level. These adjunct staff have much to contribute to a university department, and their involvement with the university will feed back to the practice of their organizations. The formalisation of the links between these scientist/practitioners and universities is a logical step in the development of the discipline.

One of the difficulties in supervising PhD candidates is that they often have no experience of the nature of information systems practice. It is difficult to help them design research that is both rigorous and relevant to industry. An internship program can help redress this problem. Students can work in industry to develop the topic for their research. Interns should be supervised by their academic supervisor and in industry by staff who hold honorary university appointments. These industry supervisors can become associate supervisors of the students once they return to the university to complete their PhD. The links between the university and the industry sponsor are likely to be closer and of much higher quality than in undergraduate sandwich and industry based learning programmes.

Most academics take study leave to conduct research and establish or renew links with other researchers. It should also be possible for academics to have a form of leave that enables them to participate in practice for a short period. This could be enabled by staff exchange or could be funded by the teaching load that can be provided by the honorary staff appointments already discussed. This form of leave can stimulate ideas for both teaching and research in ways that are not possible with other forms of engagement.

Conclusions

Information systems has forms of professional engagement that vary in quality and impact and can often be considered superficial and conservative. Professional engagement needs to be significantly improved, especially in terms of the movement of staff and students across the industry/university boundary, in order for the discipline to mature.

The development of forms of professional engagement which adequately address the need to initiate students into the methods of professional practice is a crucial component of an effective information systems curriculum. There is still considerable uncertainty within the information systems discipline about what constitutes the elements of professional practice and what methods of teaching are best suited to each element. We do not believe that any one method is capable of meeting all aspects of the problem. The task confronting information systems educators in this area has three key components:

(i) to define more precisely the content required of a professional practice component to the curriculum. In particular it is important to explore further the aspects of professional practice which go beyond simple description of professional skills and responsibilities, and to incorporate the teaching of the patterns of behaviour and action used by successful practitioners.

(ii) to establish a clearer picture of the strengths and weaknesses of the measures such as those described in this paper which are currently used as a means of addressing the need to teach the different aspects of professional engagement.

(iii) to adopt some of the engagement methods of the major professions.

References

Brancheau, J.C.; Janz, B.D. Bc Wetherbe, J.C. (1996) "Key Issues in Information Systems Management 1994-95 SIM Delphi Results", *MIS Quarterly,* 20:2, pp 225-242.

Broadbent, M.; Butler, C.; Hansell, A. ~ Dampney, CNG. (1995) "Business Value, Quality and Partnerships: Australasian Information Systems Management Issues", *Australian Computer Journal,* 27:1, pp 17-26.

Glazer, N. (1974) "Schools of the Minor Professions", *Minerva.* 12:3, pp 346-363

Hughes, E. (1959) "The Study of Occupations" in Merton, R.K., Broom, L. and Cottrell, L.S. (eds) *Sociology Today,* Basic Books.

Illich, I. (1970) *A Celebration of Awareness: A Call for Institutional Revolution,* Doubleday.

Keen, P.G.W. (1987) "MIS Research: current status, trends and needs" in Buckingham, R.A.,Hirschheim, R.A., Land, F.F. and Tully C.J. (1987) *Information Systems Education. Recommendations and Implementation,* British Computer Society Monographs on Informatics, Cambridge University Press.

Klobas, J.E. & McGill, T. (1993) "Computing Professionals and Information About Developments in Information Technology", *Australian Computer Journal,* 25:4, pp 149-158.

Larson, M (1979) "Professionalism: Rise and Fall", *International Journal of Health Services,* 9:4.

Lee, D.M.S.; Trauth, E.M. ~ Farwell, D. (1995) "Critical Skills and Knowledge Requirements of IS Professionals: A Joint Academic/Industry Investigation", *MIS Quarterly* 19:3, pp 313-340.

Moore, W. (1970) *The Professions,* Russell Sage Foundation.

Schein, E. (1973) *Professional Education,* McGraw-Hill.

Schon, D.A. (1987) *Educating the Reflective Practitioner,* Jossey-Bass.

Schon, D.A. (1991) *The Reflective Practitioner,* Arena.

Simon, H. (1972) *The Sciences of the Artificial,* MIT Press.

Watson, R.T. (1989) "Key Issues in Information Systems Management: An Australian Perspective", *Australian Computer Journal,* 21:3, pp 118-129.

Development of a *Masters of Science in Information Systems* Program Aligned With the ACM/AIS/AITP IS'97 Paradigm: A Presentation/Panel Discussion

Herbert E. Longenecker, Jr.: bart@cis.usouthal.edu
Roy J. Daigle: daigle@cis.usouthal.edu
School of Computer and Information Science
University of South Alabama, Mobile AL 36688

Abstract

Within the context of a move from the quarter to semester system, and with the release of the IS'97 curriculum model, we wish to share the opportunities we find to develop an extension to the IS'97 model which incorporates our varied groups seeking the Masters of Science in Information Systems Degree. We show paths for 1) IS undergraduate majors, a 2) Five Year fast-track for excellent students, and a path for those with 3) non-IS undergraduate degrees. The heart of our program addresses the major design goals of IS'97 to develop skills in A) Development of Information Systems, and B) Deployment of Information Systems utilizing current information technology.

Introduction

Information Systems curricula were introduced in the early 1980's by DPMA and ACM. The DPMA curriculum was revised in 1986. The ACM 1982 model curriculum used an almost identical model for both graduate and undergraduate programs. Differences between graduate and undergraduate programs involved the degree of learning. We followed the ACM 1982 model at our institution and have taught courses with almost identical titles to the ACM document. In many cases we have even cross-listed graduate and undergraduate courses. However, courses in Information Analysis and Design, and Database have always maintained a distinctive flavor within each program.

Prerequisite Undergraduate Experience

We have observed that students with inadequate preparation in programming do not do well in our advanced graduate courses. In fact, no student who has received a C grade in these beginning courses has completed our graduate program [Doran, 1994]. Therefore, we have designed a CS 1,2 sequence similar in scope to IS'97.5 within the prerequisite sequence.

We feel that fundamental knowledge of computer structures, operating systems, and telecommunications are necessary prerequisite material for students who will be involved in Information systems analysis and design course. While students can participate without the associated skills, their success is limited since they do not have the experience needed to recognize many opportunities which are components in realistic solutions. Yet, we do not feel that advanced courses involving these topics are prerequisite to successful study at the graduate level.

IS'97.9 is a course that utilizes programming in a database environment. In our experience we find that many of the skills presented in IS'97.2 as well as IS'97.9 can be taught as an extension to the programming classes, and as a prerequisite to *Enterprise Database Modeling* and *IS Analysis and Design.* This same course can be used as a capstone experience for our undergraduates, while being prerequisite for the graduate experience. See Figure 1.

In Figure 1, the relationships between graduate and undergraduate programs are evident. The majority of our students are non-CIS degree students. Thus, the prerequisite (to the graduate program) courses are required by most graduate students.

Many non-CIS degree students also require courses in accounting, finance, management, and sometimes PC tools, math, and/or statistics. In our new program, students entering in the fall can complete the entire prerequisite experience in one calendar year.

The *Database Programming* course is an extremely intense summer course. The *OO Programming, Data and File Structure*, and the *OS/Telecommunications* courses are taught in parallel over two semesters.

Master of Science in IS Courses

Two major thrusts of IS'97 are development and deployment of information systems. These concepts are the two exit curriculum areas of IS'97. These areas represent the underlying sets of principles of our MS degree program. IS development concepts are initiated in the *IS in Organizations* (see Figure 1) course and expanded with the *DBMS Modeling*, *Information Analysis*, *IS Design*, and *IS Project* sequence. Deployment issues are addressed in the *Data Warehousing* and *DBMS Administration* courses, and then expanded with selections of elective courses.

IS in Organizations sets the context for the entire program. The role of this course was identified by Nunamaker et al 1982, and was preserved and expanded in IS'97.3. The strategic role of IS in applying information technology to optimizing solutions of the problems of people is a central theme. Systems theory, business process development, and strategic project management are important targeted areas of emphasis. Concepts of personal decision making, ethics, and professionalism are also developed.

DBMS Modeling builds on the foundation established in the Database Programming course. ER modeling is expanded through the use of DBMS schema development tools. Access and then tools such as ERWIN or S-Designer are used to take advantage of desk-top as well as enterprise modeling tools. Event driven programming is practiced. A significant amount of time in data modeling for report generation.

The *Information Analysis and Design* sequence involves learning and then applying project management as well as the steps involved in a complete methodology with emphasis on audit and control concepts for IS development. "Thin" and full-client server approaches are used for enterprise level problems. Rigorous IS development projects are attempted and completed within the follow-up IS Project two semester sequence. Students work in teams, but will have individual accountability at each level for life cycle tasks.

Data Warehousing and DBMS administration courses will allow students time to explore significant issues involving the deployment, including installation and management, of an enterprise level system. Advanced electives will allow students to specialize in complex networks and web site management and programming.

As an alternative to completing an IS project, students may take part in thesis research. A full year is recommended for initiating and finishing a thesis. The content areas for thesis research may include the pragmatic aspects of the IS discipline as well as more theoretic areas.

Alignment of this MS Program with IS'97

The courses of our MS program are not the same as those within IS'97. However, we believe that the spirit of IS'97 was more to facilitate concepts than to use the specific IS'97 courses. The learning units, however, serve as specific check-points. In Figure 1 we have indicated rough course equivalence to the IS'97 courses. These equivalences have enabled us to match the learning units of IS'97 to our courses. The mapping of these learning units, combined with statements about our interpretation of the learning units -- we call such statements, *local objectives* -- make it possible for a 100% mapping of IS'97 learning units to our courses. Our learning unit local objectives also allow us to express the levels of depth we expect for graduate students, and therefore, form the basis of exam questions.

References

Davis, Gordon B., John T. Gorgone, J. Daniel Couger, David L. Feinstein, and Herbert E. Longenecker, Jr. 1997. IS'97: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. Published by AITP, Chicago 1997.

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.

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.

Doran, Michael V., David Langan, 1994. School of CIS, University of South Alabama, internal study.

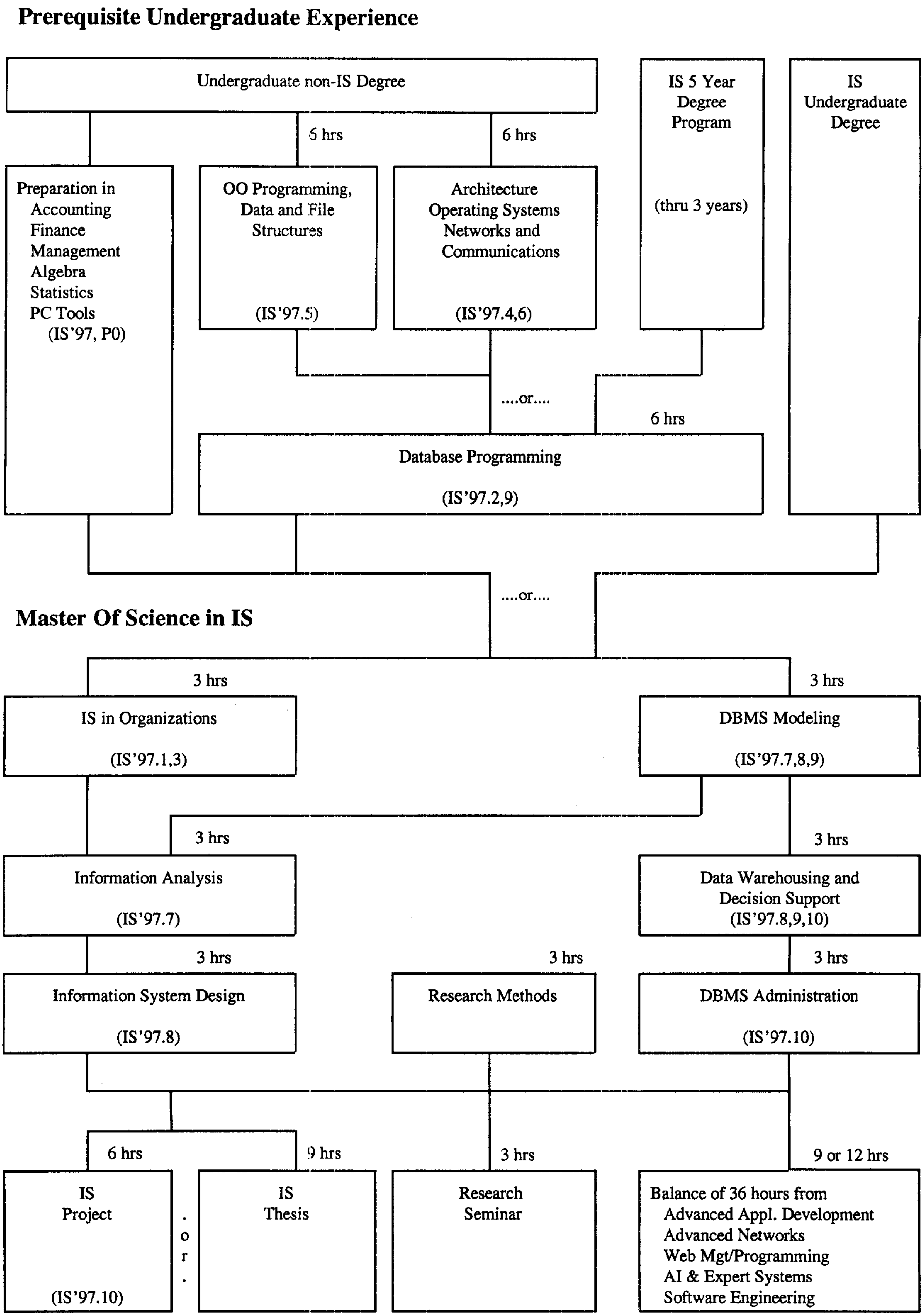


Figure 1. IS'97 Compliant Masters of Science in Information Systems.

Integrating Information Systems Ideas into the Computer Science Curriculum - A Case of Two African Universities

Dr. Sunday O. Ojo
Department of Computer Science
University of Botswana.
Private Bag 0022,
Gaborone, Botswana
Fax: 267-3552784
Email: ojoso@noka.ub.bw

ABSTRACT

This paper presents the author's experience in integrating Information Systems ideas into Computer Science curricula in two African Universities. The aim was to balance usability and computability concerns in computing. A number of socio-political and pedagogical issues that emanated from the two cases are discussed. The paper advocates for the evolution of a more flexible and comprehensive paradigmatic framework that treats the computability concern and usability concern as two sides of the same coin in the development of any computing education programme.

1. INTRODUCTION

1.1 The Problem

The disciplines of Computer Science(CS), and Information Systems(IS) taken together, represent much of academia's coverage of the use of computers and the creation of computing tools and techniques to solve problems. It is the failure of CS as a discipline to address practice-oriented issues that has resulted in the various quests to create other computing disciplines, notably, IS and Software engineering[Denning et al, 1989]. In practice, this split encourages these disciplines to diverge while defining their own territories. A consequence of the divergence of these disciplines is that important aspects of computing will be omitted from study in academic curricula [Wilson, Greenleaf & Trenary, 1989].

Broadly speaking, we can identify two forms of concern in computing which are supposed to be addressed in the quest to establish a computing discipline such as CS and IS. That is computability concern and usability concern [Kling, 1996].

Basically, computability concern(CC) deals with formulating mathematical and scientific theories of computing and providing formalised tools and techniques with the controlling objective being primarily to ensure technical impeccability of the products of computing activities - e.g. technical efficiency, portability, completeness, and other technical performance issues of computing systems. It also focuses on the theoretical soundness of the underlying ideas. Usability concern (UC), on the other hand, deals with giving adequate recognition to what we shall call the "contextual milieu" of computer applications and evolving appropriate mechanism to handle them. By contextual milieu we mean the human and organisational related issues including social, economic, cultural, organisational issues of computing in practice.

Traditional perception of the CS discipline, as can be seen in most of the proposals for CS curriculum to-date is that motivated by CC and so most CS curricula are CC-dominant. The more recent of these proposals [Denning et al, 1989; Tucker et al, 1991] seem to give some level of recognition to the need for treatment of UC. There is the common perception of CS as being purely a mathematico-scientific discipline. This engenders the doctrinaire belief in the purity of CS, which should not be touched by social facts of reality which exists is UC. From the experience in the two cases considered in this paper, this conception of CS often leads to a mismatch between theory and practice, which in turn results in a mismatch between the graduates of CS programs and market demands. This constitutes a limiting factor in ensuring the development of appropriate indigenous human resources capacity for effective deployment of the computing technology in these countries. Consequently, achieving the socio-economic development objectives for adopting the technology remains a mirage.

A study carried out by two students[Familusi &

Ademola, 1991] under this author's supervision, revealed that most of the graduates of the Ibadan programme are more competent in computability-related (technical) aspects of computing but are somehow handicapped in usability-related(non-technical) aspects - e.g. information systems requirement analysis, social analysis of computerization, human communication and management aspects. This is a key reason why the project reported in Ojo(1992) was undertaken.

The report of a consultant on CS education in Botswana[Muller 1993] observed that the initial CS curriculum did not cater adequately for producing graduates for computing practice in the industry. It further noted that there is a need to refocus the CS program such that it addresses the specific needs of Botswana so as to obtain sustained support and to have a program of size, scope, and quality which is relevant to the needs of Botswana and supports the effective use of IT. This report was a trigger for reviewing the CS curriculum. It was during this review exercise that most of the IS ideas were integrated into the curriculum.

1.2 Paper Objective

This paper presents the author's experience in integrating Information Systems ideas into Computer Science curricula in two African Universities - the University of Ibadan in Nigeria and the University of Botswana in Botswana. The goal of the integration in the two universities was to balance computability and usability concerns of computing. A number of socio-political and pedagogical issues that emanated from the two cases are discussed. It is a view held in this paper that there is a need to make appropriate balancing of CC and UC in any computing education programme. It is stressed further that both CC and UC are part of the same process, and therefore should not be viewed as competing alternative processes of thought, rather, they are successive and complimentary episodes of thought involved in computing education programme. Following the view in Ogunniyi (1986), CS as a scientific discipline, should be seen as a truly dynamic human enterprise rather than a dogma. Therefore, it is advocated in this paper that a more comprehensive and flexible paradigmatic framework for the discipline of CS which adequately addresses these two concerns should be promoted.

2. CS CURRICULA AT IBADAN AND BOTSWANA

2.1 The Two Contexts in Comparison and Contrast

2.1.1 Academic Structure and Environment

The Ibadan and Botswana CS programs were started in 1974 and 1992 respectively. They are both located in the Faculty of Sciences of the respective universities. Ibadan and Botswana operate a Course system and a Subject system, respectively. With the Ibadan course system, there is curricular rationalisation university-wide in which case, duplication of courses across departments are discouraged. As a result, the system supports, say, CS students, taking courses offered in some other departments, including science, humanities, social sciences, etc., in addition to the minimum compulsory courses in CS. In the Botswana case, the Subject system adopted seems to encourage each department to be more or less self-sufficient for the course modules required for each academic program. In the Botswana CS program for instance, all the courses required to be done by the students are contained within the department curriculum. Hence, any non-traditional CS ideas such as those found in IS, can come into the syllabus as course modules or integrated into existing course modules as topics.

2.1.2 Entry Requirements and Curricular Goals

The two curricula have similar goals, which are, to produce graduates with academic competence that enables them to pursue academic career in CS, as well as professional competence to function effectively in the industry. The students background and entry requirements are generally the same. Prospective students are expected to have background in science and mathematics as prerequisites for admission into the programs. The first of the four-year duration for the programs is mainly devoted to equipping students with requisite science and mathematics knowledge.

2.1.3 Faculty Members

With respect to faculty members, they are more international in the Botswana case than in the Ibadan case which is predominantly local-staffed. However, common to the two cases is the fact that the faculty members are dominantly CC-oriented, by training and experience. The result is that there are always more voice for CC-dominant CS curriculum than those advocating for a place for UC in the curriculum. The UC group usually has a hard time convincing the CC group of their view.

2.1.4 Curriculum Development Process

Curriculum development process in the two cases are similar. The department initiates the process by coming up with a proposal. The proposal goes through somehow bureaucratic approval processes both at the Faculty board and University Senate levels. A key difference in this process is that the system in Botswana encourages more stakeholder participation than that of Ibadan. This situation at Ibadan makes it more conducive for the curriculum to be dominated by the interests and views of a minority stakeholder, as dictated by their background and pedagogical orientation. As for the Botswana case, with encouragement of full participation of all stakeholders, reaching a consensus is often an up-hill task, hence, a curriculum could be a product of political process dominated by compromises.

2.1.5 External Input in the Curricula

In both Ibadan and Botswana, provision is made for industrial attachment (or internship), during the summer long vacation. At Ibadan, this is made optional, while it is compulsory at Botswana. Some feedback based on the performance of the students is obtained through this interaction with the industry. In contrast with Ibadan, there is in Botswana, a provision for an Advisory Board for each academic program. Such a board is composed of representatives from the industry and government, together with CS faculty members. Its main duty is to ensure that plans and priorities of the department are realistic and relevant. However, hitherto, getting such a board for the CS program has not materialized. So, in both cases, besides the student industrial attachment program, not much of industry-academia collaboration is going on. This, to this author, is a source of apparent pedagogical conservatism in the proponents of a CC-dominant CS curriculum.

2.2 Approach to Integrating IS Idea into the CS Curricula

Options opened for use in the integration process include: a) integrating IS ideas as course modules into the syllabi; b) incorporating IS topics into existing CS course modules; c) identifying suitable course modules in other departments as options for CS students, and d) introducing a specialist IS program at the graduate level for CS graduates. The settings in each of the two universities as summarised in section 2.1 above, had a bearing on what choice of options (or combination of options) was made.

2.2.1 The Ibadan Approach

Options (b), (c) and (d) were used at Ibadan. In using option (b) IS topics were incorporated into a number of course modules including Introduction to CS, Systems Analysis, and Computer Applications. Such topics were aimed at addressing social, cultural, organisational and management issues of computerization. However, this option did not go far enough in addressing UC. With option (c), course modules in the Faculties of Social Sciences and Humanities were identified as options for CS students. E.g., among the popular courses in the Social Sciences were Accounting(Financial and Management) and Operations Research from Economics department. The Course System adopted at Ibadan greatly facilitated the use of this option. An attempt was made to use option (d) through a proposal developed by this author as documented in Ojo(1992). This proposal was for a Masters in Information Systems programme for graduates of CS programs of the universities in the country. It was mainly aimed at addressing the problem of the academia-industry and theory-practice mismatch in computing competencies of the graduates. However, this proposal could not be pursued further as the author was "brain-drained" out of the country not long after making the proposal.

2.2.2 The Botswana Approach

The review of the first draft of the CS curriculum which was developed by the Mathematics department, gave an opportunity to integrate some UC-driven IS topics and course modules into the curriculum. More integration of IS ideas into the curriculum was done at a later curriculum review exercise.

Based on a submission made by this author to the Curriculum Review Committee members[Ojo, 1992], it was agreed that in addition to equipping students for graduate studies in Computer Science, the curriculum should

i) be aimed at producing graduates who are adequately equipped to effectively function in various positions of responsibility for organisational computing systems deployment involving defining and planning, eliciting requirements for, designing, implementing and managing development and operations of, computer based ISs;

ii) stress the need for organisational knowledge including understanding of (a)organisations, organisational processes and functions within organisations; (b) socio-political problems commonly encountered in the computing application systems development process and management skills for handling them; (c) the common concepts, strategies and tools required of an IS analyst to effectively play his multi-faceted role; (d) human relations and interpersonal skills for communicating effectively with the other actors in systems development projects.

This served as a basis for the statement of the driving philosophy and goals for the curriculum. It also became a reference point later in the process, to argue for integration of IS ideas into the curriculum. However, as each of the members of the committee had their own prejudices and biases, dictated by their varied academic experiences and background, agreement on how to fashion the curriculum in a way that would ensure appropriate CC-UC balancing turned out to be a difficult one to reach. At the end, a notable change to the curriculum was an appreciable expansion of the horizon of the UC in the curriculum.

The Subject system adopted by the university dictated that only Options (a) and (b) were used. In using Option (a) IS topics were incorporated into a number of course modules. For example, Systems Analysis and design which originally took a purely technical orientation, was refocused to take a socio-technical perspective; into another course module titled Software Packages (a course intended for teaching productivity software packages such as Wordprocessing, Spreadsheet, etc) was incorporated topics on the underlying philosophy and applications requirements of DSS, Office Information Systems, etc. In using Option (b) new course modules were introduced. These include Communication Skills, Information Systems & Organisations, and Management, Social Issues of IT, and Economics of IT, as well as a second course module on databases, partly aimed at exposing students to more management and organisational issues of database systems.

Not every faculty member totally agreed to having courses such as these in a CS curriculum. So, getting the courses into the curriculum involved compromises. All, of these new course modules, except the Communications Skills and IS and Organisations, were made optional courses. Also, in the selection of courses that will be taught in a year, attempts are often made to push some of these IS courses aside for the year. So, each year, there is always a battle to be fought on this ground.

3. KEY ISSUES EMANATING FROM THE TWO CASES

3.1 Teaching the IS Course Modules

In both the Ibadan and Botswana cases, hitherto, most of the faculty members in the department do not have competence in teaching the IS course modules introduced. When it comes to recruitment, candidates without pure CS higher degree do not have much chance of being recruited. Some of the courses are being taught by few interested faculty members in the depart and some others from outside the department. For example, in Botswana, Management is being taught by a faculty member from the School of Accounting and Management of the University, and Communications Skills by faculty members from the Communications and Study Skills Unit of the Faculty of Science. Since most of these courses are made optional, anyone of them which no one is available to teach each year is easily pushed aside for the year. CS faculty members teaching some of these courses do not have background education in the areas, rather, they draw from their experience in their involvement in organisational systems development projects. Also, it remains a problem, getting the non-CS faculty members to orient the teaching of the courses to the particular needs of CS graduates. The critical issue of who would teach the IS courses in the CS programs remain unresolved.

3.2 Socio-Politics of CS Curriculum Development.

Experience in these cases revealed that, the path of a curriculum development process could be strewn with socio-political considerations, rather than pure rationality. For instance, a curriculum could become fashioned to suit a particular individual or group interests; another department may seek to influence the curriculum content to ensure its indispensability in the implementation; budding research ideas that have not been subject to peer assessment nor widely accepted could find its way into a curriculum; course modules may reflect what an individual wants to teach, rather than what should be taught, and so on. Also, it is not uncommon for faculty members to adopt the doctrine of purity of CS, as a tool and symbol of established interests, power, and authority; and ultimately used as a barrier to appropriate balancing of CC with UC in a CS curriculum.

3.3 The Doctrine of Purity of CS.

The shaping of the CS curricula in the two universities involve faculty members of diverse views on the necessity for balancing CC and UC in a CS curriculum. Some faculty members hold a doctrinaire belief in the purity of CS, which they would argue that it should not be "adulterated" with the "soft ideas" of IS. This group would dub anything IS-related as being "soft sciences" which have no place in the curriculum of CS. Some others have argued that science and technology is value-free, and so bringing in any social, cultural, or organisational issues of IS(which are inherently value-laden) is a negation of this "universally accepted" view of science. This was a major source of contention in the quest to integrate IS ideas into the CS curricula.

3.4 Goal-driven Curriculum Development.

In the two cases, the driving philosophy and goals of the programmes were aimed at producing CS graduates with potential for productive computing practice in the industry, and productive scholarship in the academia. The experience revealed that it is one thing to make a statement of driving philosophy and goals, it is another to drive the implementation of the curriculum in this direction. Where faculty members are divided along the computing-as-knowledge and computing-as-practice dichotomy, the usual turf-battle would rage. This situation makes it all the more difficult(if not impossible), the realisation of the dual goal of ensuring academic and industry career competencies through paradigmatic integration of IS with CS.

3.5 The publish or perish syndrome

The well-known publish or perish syndrome in most of our universities is another key issue that has a bearing on balancing CC with UC in a CS curriculum. CC-dominant research projects do thrive better in comparison with UC-dominant ones. This is so because, most of the former are laboratory-based and are commonly limited to the realm of computer systems with a higher chance of producing publishable results early. On the other hand is the UC-dominant research which is usually organisational- or societal-based. By its nature, this is subject to human factors, that may make it to have lower rate of returns. It should not be surprising therefore that a typical computer scientist would opt for CC-dominant research projects where he can make up the quantity requirement of the publish or perish syndrome. By this, most IS ideas which would have been encountered through UC-oriented research endeavour, are missing in the intellectual culture of the computer scientists. This has a far-reaching implication in balancing CC and UC in a curriculum. This is so because of the fact that as research ideas become developed and accepted they would have impact on courses which provide basic training[Wilson, Greenleaf & Trenary, 1989]. By this, CC-dominant ideas would continue to shape the direction of developments in the CS discipline.

3.6 Curriculum Innovation

The need for curriculum innovation is a well-established belief among educationists. It is commonly believed that education does not take place in a cultural vacuum; every teaching and learning has a geographical, a historical and a socio-economic context [Wilson, 1981]. However, computing curriculum innovation has not been a common practice in the two CS departments. Carrying out local needs analysis on which curriculum objectives formulation and determination of the curriculum contents will be based, is a commonly neglected aspect of curriculum development in the two cases. It is the common approach for the curriculum developers in the two cases to depend almost entirely on CS curricula developed elsewhere, as a source of inspiration. This is done without considering whether or not the contexts are similar. It is not uncommon to accuse any faculty member advocating for curriculum innovation, of wanting to complicate issues. The lack of recognition of a need for curriculum innovation was a factor that made agreement on the integration of IS idea into CS curriculum to be difficult to reach.

3.7 Providing the "Right" Academic Environment".

CS is often associated with science and science-related academic environments in most of our tertiary institutions. This is the case with the two CS programs under consideration. The result of this is that the paradigmatic framework for CS programs often gets rooted into the Mathematics and pure Sciences. Undoubtedly, the fact that the two CS programs are accommodated in this kind of academic environment, was a limiting factor with regards to which IS ideas were integrated into the two CS programs. In a traditional "scientific" environment, it is an up-hill task, circumventing the traditional intellectual culture of our academic planners such that the required paradigmatic framework gains intellectual feasibility. Is what is needed paradigmatic separation, accommodation, integration, or co-existence? Whatever the answer may be, providing the right academic environment constitutes a key pedagogical challenge.

3.8 Stakeholders Participation

The educationist [e.g. Wilson, 1981, page 8] have suggested that where more employers are involved with those in the education system in defining these broad purposes of an education and training programme, a clearer goal is established. At Ibadan, there was no provision for the setting up an advisory committee of stakeholders. In the case of Botswana, the university has a provision for having an Advisory Board for each program. However, this avenue has never been used to get inputs into the CS curriculum from stakeholders outside the academia. Since most of the UC issues originate from outside the academia, the best source of knowledge of these issues are those stakeholders outside the academia. Excluding these stakeholders from contributing to a computing education programme development would therefore mean that the academia would insist on a CC-dominant curriculum. Some people have argued that involving non-academic stakeholders in curriculum development could result in "unacademic ideas" being introduced into the curriculum. While every care has to be taken to avoid undue dictation from outside the academia about what is(not) to be taught in an academic program, the vital role that these non-academic stakeholders could play in making computing education program relevant to the local context, should not be ignored. It is a key issue emanating from the experience in the two cases, that there is a need for evolving appropriate mechanism for involving the non-academic stakeholders in computing education programme development. Their involvement could provide an arbitrating forum for the combatants in the turf wars on balancing CC with UC in a CS curriculum.

Forging appropriate academia-industry link could facilitate stakeholders participation too. However, it has been an up-hill task, convincing the people in the industry that they are partners with those in the academia in ensuring that appropriate CS graduates are produced. Also, it is rare to have a research project funded by the industry. No formalised mechanism has yet been established for obtaining information feedback on the industry assessment of graduates of the programs. Generally, evolving a mutually beneficial link with the industry remains a key issue which has to be resolved. The lack of it is a major factor militating against coming to terms with the need for balancing UC with CC in CS curricula.

4. CONCLUSIONS

In this paper we have presented a case of CS programs in two African universities in which attempts were made at balancing CC and UC by integrating IS ideas into the CS curricula. Given the various socio-political and pedagogical issues emanating from the experience in the two cases, it is predictable that the battle line between the "pure computer scientists" and those advocating for IS ideas in CS curriculum would continue for quite some time to come. However, given the situation in some universities, such as those in South African whereby CS and IS coexist as academic programs within the same department, there is a cause for optimism about CS and IS growing together in African universities.

The socio-economic context of the computing applications in Africa raises problems of a new sort that require considerable restructuring of the traditional conception of CS as a pure science. It requires not just simply applying ready-made mathematico-scientific paradigms to new situations; rather finding new paradigms, or new ways of interpreting the old paradigms in order to appropriately address the challenges presented in the deployment of IT in Africa. This definitely requires an appropriate integration of "hard science" and "soft science" paradigms into a single whole, to evolve a more flexible and comprehensive paradigmatic framework that gives equal treatment to computability and usability concern in the development of any computing education program. Evolving such a framework does pose an enormous challenge in the light of the issues that emanated from the two cases considered in this paper.

REFERENCES

Denning P.J. et al.(1989), Computing as a Discipline, CACM Jan.

Familusi B.F. and Ademola O.E.(1991), An Assessment of Utilization and Impact of Computer Technology in the Finance Sector of the Nigerian Economy. Student Computer Applications Research Project Report; Dept of Computer Science, University of Ibadan.

Gibbs N.E. & Tucker A.B.(1989), A Model Curriculum for a Liberal Arts Degree in Computer Science, IEEE Computers. Sept.

Kling R.(1996), Computerization and Controversy - Value Conflict and Social Choices. 2nd Edition, Academic Press. pp.32-38.

Muller M.E.(1993): Report on Computer Science Education and Opportunities for Information Technology in Botswana. A Consultancy Report.

Ogunniyi M.B.(1986), Teaching Science in Africa. Adebara Press, Ibadan.

Ojo S.O.(1992): Suggestions for a Review of the Proposed Computer Science Curriculum. An Internal Communication, Dept of Mathematics, University of Botswana.

Ojo, S.O.(1992): Curriculum for a Masters Programme in Information Systems for Nigerian Universities. In Bathnagar, S.C.(ed), IT Manpower: Key Issues for Developing Countries, Tata McGraw Hill, New Delhi.

Tucker A.B. et al.(1991), Curricula 1991, A report of the ACM/IEEE-CS Joint Curriculum Task Force, IEEE Computer Society Press.

Wilson B.(1981), Cultural Contexts of Science and Mathematics Education: A Bibliographic Guide. Centre for Studies in Science Education, University of Leeds, Arthur Wigley & Sons Ltd.

Wilson J,Greenleaf Newcomb, & Trenary R.(1989); Algorithms and Software: Integrating Theory and Practice in Undergraduate Computer Science Curriculum.

Enterprise Oriented Curriculum for Information Specialists

Doris Lidtke
Towson University

Jimmie E. Haines
The Boeing Company

David L. Feinstein
University of South Alabama

The panel will discuss the new curriculum developed to meet the Profile of the Graduate which was developed by the industry members of the Task Force. The Panel members will emphasize the ways in which this curriculum is different from current curricula, how the graduate are better prepared to meet the need of industry, and how the courses/modules can be sequenced. The project has been funded by NSF/DUE.

I Am Scheduled to Teach "World Wide Web"...Now What? An Examination of Curriculum Development for WWW Courses

J. Douglas Robertson
William T. Schiano
Department of Computer Information Systems
Bentley College

Abstract

The World Wide Web is being widely diffused across college campuses. This paper details the design and delivery of an introductory undergraduate World Wide Web course. The motivation and philosophy of the course are provided, along with descriptions of each week's content and assignments. Lessons for instructors are extrapolated from experiences in the delivery of the course. Finally, the paper explores the current and future role of World Wide Web courses in the curriculum in light of the technology's rapid evolution. Appendices include the course syllabus and a list of useful URLs.

Importance of the Word Wide Web.

In newspapers, in magazines, on radio and television, URLs beckon us to access information and explore the World Wide Web (Web). Very quickly the Web is taking a prominent place in our repertoire of tools to acquire and process information. Today's students need a fundamental understanding of this technology's capabilities and limitations. The offering of an introductory Web course is necessary to give students control over a technology which will become an integral part of their education (Thomson and Goyal 1996).

This paper describes our experiences in designing and delivering a 1.5 credit course on the Web. The first section describes the motivation for the course, followed by a discussion of the content of the course itself. We then step back to examine the lessons we have gleaned from the experience and offer some insight into the evolving role of the WWW in undergraduate curricula.

The Course: What we are trying to do.

The course is open to all students who have completed the Introduction to Information Technology course required of all students in their first semester that will provide them with a strong background in the Web. The focus of the course is on the technology of the Web and the tools available for its use and exploitation, rather than the strategic business impact, which is left to later electives. The course is hands-on, with students gaining their knowledge and understanding of the Web through exploration and review of what others have accomplished. They will also create, manufacture, and maintain their own Web presence. Our philosophy is oriented toward student success, with emphasis on looking, reflecting, and doing. We purposely decided to use no textbook for the course, opting to use the Web as its own reference materials. Numerous links are made available for the students to view and explore material supplementary to classroom explanation.

The Course: What we are doing.

The course syllabus appears in Appendix A. **Week one** of the course deals with fundamental explanation of the World Wide Web and its relation to the Internet. The concept of a browser is introduced and Netscape Navigator is used to explore the breadth of the Web. Search engines are investigated and the first assignment requires the use of one of the numerous engines available to locate and save a graphics file to the student's work disk. HTML is revealed by browsing other Web pages and examining the code. A second assignment is made for the student to find the URL of a Web page using a particular HTML feature.

In **week two**, the student gains knowledge of HTML document structure and is introduced to some of the fundamental HTML tags. Even though HTML editors are mentioned, a simple text editor, like NotePad, is used to create HTML documents. The technique of toggling between text editor and browser is stressed so that students can see the relationship between the HTML tags they use to markup the text document and the way in which a browser interprets the tags. Students learn what the tags do and what syntax is necessary for the tags to be rendered correctly. At this point, they can create their own home page, view, and test it locally. An FTP client is introduced so that students can upload their home page to an account created for them on our department server. Their next assignment is to use the FTP client to upload their own home page and associated graphics files to the server establishing their Web presence.

By **week three**, the student has mastered the fundamental HTML tags and has been apprised of some of the numerous tutorials available on the Web explaining

HTML. We begin to delve into some of the "advanced" features available in HTML. Since students are familiar with tables from using spreadsheets, showing how this powerful concept is realized and utilized in HTML is a natural next step. The Web is replete with example sites that use tables in a myriad of ways from simple collections of data to complicated design layouts. The students are able to design a Web page that uses a table to effectively communicate information and that becomes their next HTML assignment.

In **week four** HTML frames are described. Again, several Web sites that make use of frames are used as examples to show how Web pages can interact to provide information in a visually pleasing and stimulating way. Frame sets and frames are described and the various ways they can be loaded with information is discussed. An HTML assignment is given to allow students the opportunity to bring together several related pages they have created and provide navigation among them.

By **week five** the student has enough information and background so that they can solve a problem using the Web and HTML. At this time a project is assigned so that they can spend the last two weeks of the course putting together a Web-based solution. While this is going on, HTML forms are described and their functionality explained. CGI and server-side programming are investigated through an assignment where an HTML document containing a student-created form is sent to a server and processed by a CGI program. The results are returned as a web page summarizing the information from the form's fields.

Week six begins with an introduction to image maps. Browsing sites on the Web reveals that many sites use image maps in one form or another. Client-side image maps are explained and an example is created to show their basic structure and action. If a student is still formulating their project structure, she/he might find the idea of an image map useful. Since the student is continuously working on the project, no further HTML assignment is necessary.

Week seven completes the course by introducing other aspects of the Web technology. JavaScript is introduced with the goal of allowing students to be able to read other people's JavaScript and to modify the code to suit their own purposes. The idea of functions, objects, events, and event handlers are introduced and couched in a way that even novice computer users can gain understanding and control from this cursory explanation. The JavaScript object model is explained to give the student the overall picture of what is really going on when a Web document is being interpreted by the browser. Discussion of the concept of cookies and their use broaches issues of privacy and security while at the same time underscoring the power of Web site development. On the last day of class, all projects have been uploaded. The class concludes with a discussion of course goals and students' perceptions of the course.

Lessons Learned and Suggestions for Faculty

From the student perspective, the course in its first incarnation appears to be successful. A survey administered in the last class meeting solicited student perceptions and opinions of the course. Most students enjoyed the course and wanted it to be a full three semester hour course. Many expressed interest in follow-up courses extending what they had already learned about Web page development. Several expressed the need for more homework assignments. When asked for personal, positive outcomes from taking the course, numerous students mentioned their ability to create a Web page. The act of creating something visible to others seemed to be a particularly enabling feature of the course.

From the faculty viewpoint, the course exceeded all expectations. Students were excited to be learning about something so current and dynamic. Most students worked much harder than was expected and generated unique home pages. Students quickly became synchronized to the Web's availability, openness, and demands of constant change. The MTV generation can indeed concentrate on something for more than ten seconds at a time.

As with any course offered for the first time, there were many things that could have been done differently. Assignments that could be easily graded were scarce. Students wanted to know how they were doing in the class, but there was no mechanism in place to provide feedback. The course did have a frames-based home page with a listing of student names in one frame, with each name linked to the student's home page which would appear in the other frame. This facilitated grading the web pages, and allowed students to see what others were doing.

In keeping with the sense of openness of the Web, we actively supported the idea that students should view what others have done and adapt what they see when formulating their web-based problem solutions. Assignments were created that forced students to personalize what they borrowed from others. The exercise of creation was what was graded, not the percentage of work traceable to others. This approach seemed successful, and was enhanced due to early discussion of basic copyright issues and concerns. A sensitivity developed by the students toward the work and effort of others allowed them to extend, not just use, the ideas of others. For an exploration of the challenges of instilling academic honesty in electronic environments, see (Connolly 1995).

Staying abreast of these trends is daunting. One of the authors had very little Web experience and was learning on-the-fly staying seconds ahead of some students. Some students had much experience using the Web and had even created home pages. It is not reasonable to expect the entire faculty to stay near the cutting edge of Web technology since the rate of change is so high, even relative to other areas of computing. Highly motivated students will often be ahead of the most conscientious faculty member, even at the freshman level. We have found it useful to have a small cohort of faculty with keen interest in the area who can informally share interesting articles and URLs. Appendix B offers a list of resources we have found particularly useful. Demand for these courses is very high, which may allow the specialization of one or two faculty in the area, which will help them stay on top of things.

The choice to not use a textbook seems to have been viewed negatively by several students. They expressed a desire for printed versions of the numerous tutorials instead of links to them. Many were printing the pages out on their slow, single-sided, inefficient personal printers. The net impact on the environment of our effort to be "paperless" may have been negative. We are now inclined to distribute two-sided hard copy of "fundamental" reading and offer URLs for "recommended" reading.

Many topics about the Web and the Internet were given short shrift as it was decided to spend more in class time on the features of HTML and JavaScript. While this was just what the students wanted, there is a concern about the depth of conceptual understanding that the students have gained.

Current and Future Role of the Web in the Curriculum

The course is already into its second offering, this time during the last seven weeks of the semester. Plans have been made to give assignments every class meeting. Half of them will be HTML-based and the others will be worksheets causing the student to write several paragraphs about World Wide Web related topics. These will be E-mailed to the instructor for evaluation. Grades will be made available weekly.

The students will be prodded to "subscribe" to HotWired Network and read Wired News regularly. As more and more students acquire knowledge about the Web and how to create a Web presence, the course will evolve to extend and build upon the general knowledge. As new technologies are introduced, they will be assimilated into this course.

Meyer and Varden (1996) described the integration of HTML instruction into CIS 101 at Pace University. Belanger and Jordan (1995) refer to this as an "advanced" stage, beyond the introductory stage of navigation and the intermediate stage of pedagogy. It is foolhardy to assume that basic instruction in HTML will be sufficient training for web development for very long. Server push models (Kelly and Wolf 1997), Java applets, animation and Virtual Reality Modeling Language (VRML) are among the emerging technologies that are becoming commonplace on web sites. We expect that basic HTML will be incorporated in future iterations of our introductory course within the next two years. This leaves the course described in this paper in perpetual flux, staying an appropriate step ahead of the prerequisite. Fortunately, such progress is a necessity given the rapid evolution of web technology. We are already considering adopting a freeware HTML editor[1] in the middle of the course, after students have become familiar with the concept of tags. However, this editor would still make the tags visible for the user. The JavaScript/VBScript portion will likely expand in future iterations, as will calls to CGI scripts and the integration of Java applets. Graphics editing and manipulation could also be expanded.

We foresee this course as a stepping stone to courses within other business disciplines that would use the web as an underlying technology. Such courses could include "Marketing on the Web," "Communication Through Hypertext" and "Electronic Commerce." Within the CIS department, we expect to build more advanced electives in Java and JavaScript, PERL, Web Security and VRML.

References

Belanger, F. and D. H. Jordan (1995). Assessment of the World Wide Web as a Teaching Tool. ISECON '95, Charlotte, North Carolina, Education Foundation of the Data Processing Management Association.

Connolly, F. W. (1995). "Intellectual Honesty in the Era of Computing." Technical Horizons in Education Journal(April).

Kelly, K. and G. Wolf (1997). "Kill Your Browser." Wired 5(3): Cover-23.

Meyer, J. and S. A. Varden (1996). Teaching Hypertext Markup Language in the First Computing Course: One University's Experience. Information Systems Education Conference, St. Louis, Missouri, Education Foundation of the Data Processing Management Association.

Authors (Forthcoming). "The Internet on $1 a Day: Resource Needs and Management for Teaching World Wide Web Development." .

[1] Netscape Gold as of this writing was free to educational users. Lightning HTML Editor is also available in a freeware version. Myriad inexpensive packages are available as well.

Thomson, N. and A. Goyal (1996). Integrating the Internet and WWW into Information Systems Curricula. Information Systems Education Conference, St. Louis, Missouri, Education Foundation of the Data Processing Management Association.

Appendix A: Course Syllabus

Instructor	JD Robertson
Office	Morison 145
Office Phone E-Mail	617.891.2974 drobertson@bentley.edu
Course Description	CS 212 World Wide Web *Prerequisite:* CS101 This course explores the World Wide Web as an educational resource.
Objectives	Upon successful completion of this 1.5-credit hour course, the student will: 1) understand the terminology associated with the Internet and WWW 2) be facile with a WWW browser 3) be able to create a WWW home page 4) understand the technological infrastructure of the Internet 5) be familiar with emerging Internet trends and technologies
Classroom	CLA413
Class Meeting Time	CS212-0S2 TTH 9:55am - 11:10am Mar 4 - Apr 29 (minus spring break) CS212-0S4 TTH 3:35pm - 4:50pm Mar 4 - Apr 29 (minus spring break)
Office Hours	TTH 1:00pm -2:15pm, TH 7:30pm - 8:45pm and by appointment
Evaluation	Activities/Quizzes 80% Project 20%
Resources	Extensive use will be made of the Bentley Information Resource (BIR) and the CIS Web Server.
Special Considerations	Activities and projects are to be accomplished by individual students. All assignments will be due on the date and time specified when the assignment is made. Any assignments turned in late will be assigned appropriate credit as determined solely by the instructor.

Date	Class	Topics	Activities assigned, goto URL: http://cis.bentley.edu/jdrobertson/
Mar 4	A	Expectations Using Netscape Navigator Search engines	A Find and save GIF file
Mar 6	B	Web page variety HTML revealed	B Find URL of web page that uses Tables
Mar 11	C	Beginning HTML	C Create personal home page
Mar 13	D	FTP HTML editors	D Upload personal home page
Mar 25	E	Client/Server architecture, TCP/IP, HTTP, server software HTML Tables	E Discuss URL
Mar 27	F	More HTML	F Create HTML document using Tables
Apr 1	G	HTML Frames	G Discuss TCP/IP
Apr 3	H	Browser wars	H Create HTML documents using Frames
Apr 8	I	HTML Forms	I Discuss Browser wars N Project
Apr 10	J	CGI Common Gateway Interface	J Create HTML document using Forms
Apr 17	K	Image Maps	K Discuss
Apr 22	L	JavaScript	L Create HTML document with Image Map
Apr 24	M	More JavaScript Cookies	M Discuss Java vs. JavaScript
Apr 29	N	Projects due and Wrap-up	

Appendix B: Useful Web Sites

1. http://junior.apk.net/~jbarta/tutor/makapage/
   Good turtorial for learning how to create a first home page. Contains links to several other useful tutorials: Tables Tutor, Frames Tutor, and Forms Tutor

2. http://www.webmonkey.com/webmonkey/teachingtool/
   Excellent introduction to HTML and the basic tags. Part of HotWired site. From this document, the student can reach information on many interesting and informative articles about the Web.

3. http://junior.apk.net/~jbarta/weblinks/index.html
   Lots of links to Web page resources: Buttons, Icons, animated GIFs, backgrounds, sounds, tools and more.

4. http://www.ipswitch.com/ws_ftp_reg.html
   Download free FTP client.

5. http://www.wired.com/news/
   Wired News provides articles of current interest as well as a searchable database of previous articles.

6. http://www.hotwired.com/frontdoor/
   Provides access to WebMonkey and its searchable archive of articles. Students can access myriad "how to" information.

7. HTML Editors
   a. Netscape Gold (http://www.netscape.com)
   b. Lightning HTML Editor (http://www.owens.cc.oh.us/Lightning/HTML_Editor/)
   c. For reviews of HTML editors, see http://homepage.interaccess.com/~cdavis/editrev/index.html

8. http://www.browserwatch.com
   A site dedicated to monitoring developments in web browsers and related technology.

OnLine, Web Based Learning Environment for an Information Systems course: Access logs, Linearity and Performance

Sheizaf Rafaeli
sheizafr@shum.huji.ac.il
and
Gilad Ravid
msravid@pluto.huji.ac.il
School of Business Administration, Hebrew University of Jerusalem

Financial support for the development of this system and for this study was provided by the Zaggagi fund, the Hebrew University of Jerusalem, and the Open University, Tel Aviv in Israel. The authors wish to express their gratitude to Ori Keren, Rachel Ben-Hanoch, Anne Yarchi-Cohen, Yona Goshen, Itamar Shabtai and Tamir Bar-Netzer for their various contributions in administering the classes and collecting the data

Abstract

This is a study of an online, Web based learning environment developed for an introductory business information systems course. The development of the environment was guided by a design principle that emphasizes choices regarding hypertextuality, centralization in client-server, interactivity, multimedia, and synchronicity. The environment included a several hundred page textbook and individual and group online learning aids and implements. Following the development period and a pilot class, the system was used by students in three large classes of 50 - 100 students each. We examined student grades, attitudes, and usage logs and their intercorrelations. Our findings indicate that online, Web-based learning environments are not just feasible. Employing such a system to complement lectures yields measurable enhancements. We propose a focus on logged, machine-collected usage statistics. Such statistics allow a measurement of actual reading behavior and linearity in the learning process. Reading amount is highly correlated with student achievement. A-synchronous conferencing tools enhance instructor-student and student-to-student interaction. This interaction is, itself, a correlate of success in the course. Furthermore, more mature student groups seem to make better use of the online environment, and use it less linearly.

Introduction

The construction and success of online learning environments are tied to the main characteristics of such computer-mediated systems: hypertextuality, interactivity, multimedia/multisense, absence of defined center (packet-switching), and elasticity of synchronicity (Newhagen and Rafaeli, 1996). In this study we examine the impacts of choices made by designers on the effectiveness of online systems in teaching information systems. We suggest a focus on access logs as an evaluation tool, and introduce the notion of nonlinearity as an important dimension in web-based browsing.

Following Rouet & Levanon, (1996) and Dee-Lucas, (1996). We are interested in the efficiency that hypertextual and interactive presentation may bring to the learning experience, as well as in the tradeoff between the power of linking and search tools and the cognitive demand or costs these tools impose on the reader

Reviews of online study environments have been made, among others, by Acker (1995), Alavi (1994), Norman (1994) Wheeler et. al. (1995) and Schutte (1997). Meta-analyses, such as those by (Kulik, 1994) and (Fletcher, 1996) also highlight the relative paucity of empirical findings to date, and the weakness of a theoretical approach guiding such design efforts. We propose to use the above mentioned characteristics in both the design and evaluation of an online learning environment for information systems. Furthermore, we propose to attach empirical measures for these characteristics, using an innovative analysis of the computer-recorded access logs.

Using access logs

Accesses to Web servers are recorded meticulously. Every request by a "client" results in a record of the date and time of the request, the transmission protocol and result status, amount of information sent, and the address of the client. If the file is protected in accordance with RFC-931, the actual username may be recorded as well. Web servers (http daemons) also keep record of errors, agents, and referrer address. See Musciano (1996) for a detailed description of record logs.

The enormous potential and rapidly growing business of advertising on Web sites have fueled an interest in marketing applications of usage logs and the information they contain. Marketers are refining the terminology and reliance on data available in these logs (Novak & Hoffman, 1996). For instance, marketing use goes beyond the raw data to employ a better distinction between hits, requests, visitors, users, identified users, and visits, and sessions. Hits are the number of page and/or graphic files requested by visitors. Requests are hits that successfully retrieve content. Visitors are individuals who visit a web site. Users are uniquely identifiable persons. An accurate count of users is made possible with additional registration or authentication. An identified user is a user for whom demographics are known and available. A visit is a sequence of requests made by one user, and a session is a series of transactions performed by a user that can tracked across successive Web sites.

In a critique of the marketing use of access logs, Stehle (1995) argues that access logs are networking-centric, not marketing centric. While many rely on the raw size of the usage log, or even the more primitive "counter", these methods are neither reliable nor valid measures of a site's quality, success or contribution. As the World-Wide-Web is "sessionless", such measures do not distinguish between "popularity" and multiple visits by one person. Nor do they identify the users or their behavior. Usage logs do not automatically distinguish between content and accompanying graphics. Dynamically generated content is not recorded. Offline reading of printed or downloaded material, mirroring, caching and proxies intervene and may bias the data. Finally, privacy and ethical considerations cf. Boehelfeld (1996), Duncan (1996) and Thomas, (1996) stand in the way of more widespread use of access logs.

Neverthelsss, measurement tools such as the evolving use of access logs in the marketing arena may be adapted for the purposes of education, in the design and evaluation of online learning systems. Empirical data collected via access logs may assist in a rational implementation of systems, and will contribute to an orderly evaluation. We intend to demonstrate one such use in the evaluation of an online learning environment here.

The Course

During the past two decades our school required an introductory, semester long, Management Information Systems course as part of the required core component of business education for both undergraduate and MBA students. The course covers a wide field of topics, ranging from definitions of information, its value and measurement, through organizational impacts, surveys of technological concepts of hardware, software, databases, data communication, artificial intelligence applications to business, legal and policy considerations, and design and analysis tools and practices. As of 1996 this course was transformed into an online learning environment, placing materials on a Web server to support ongoing regular lectures. Several motivations guided the development of this system: (1) **Language**. English is a foreign language for most of our students. The only available textbooks in the field that are up-to-date are in English. Creating Web-based learning environments allows us to create a native language version for the fraction of the cost and time required in making textbooks. (2) **Flexibility**. The subject matter changes frequently, rapidly outdating most textbooks. A two year old textbook is not likely to mention the MMX processor, discuss the Java language, report accurately on legislative changes, or emphasize the centrality of client server technology. (3) **Availability**. Textbooks are very expensive. With new editions each year, a $70 textbook has little resale value, hence students were less likely to purchase their own. (4) **Portability.** A Web-based learning environment enables use at a distance, even when the traditional weekly meeting structure is preserved. (5) **Interaction.** Opportunities for interaction are actually (and paradoxically) enhanced in a-synchronous online environments. Not only can interaction take place around the clock and all week, and is no longer limited to class and office hours. Interaction is also less encumbered by the co-presence of dozens of other students. Online learning environments also make possible the highlighting of the cognitive interaction between various concepts in the subject matter, and student-to-student interaction, a hallmark of small groups that is largely unavailable in the standard, huge introductory core classes. (6) **Cost.** Publishing in an online environment is less expensive than the production of a regular textbook, while allowing many more illustrations, use of color and animation, etc. Furthermore, the environment shell can now be used for other courses. (7) **Research.** However, we were primarily interested in developing this environment for the exciting research potentials made available through it. Surely, lessons learned in an environment of this nature are applicable to other uses of online systems such as marketing, politics, and management.

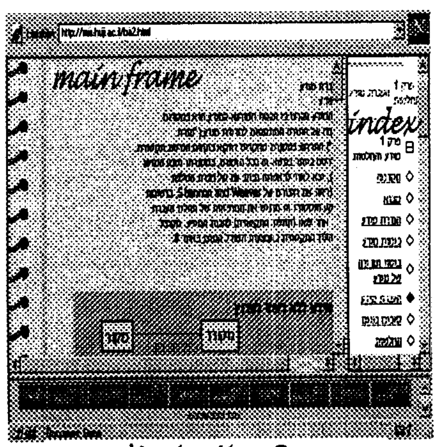
The introductory course used the online system as an accompanying textbook and between-meetings forum. It was offered four times during the last year. There was one "pilot" class of 80 students which used the system as it was built, and three classes which used a fully implemented system. The following data pertain to the these three classes: An undergraduate (BA) class, 120 students; a graduate (MBA) class, 80 students; and an "Executive MBA" class, 40 students. Table 1 displays the formats and class components for each of these classes.

TABLE 1: Classes and uses

	BA	MBA	EMBA
Lab lessons	1 hr/wk	none	none
Lecture	2 hr/wk	3 hr/wk	3 hr/wk
Semester length (weeks)	14	14	8

System design and components

The online environment was a frame-based WWW system, allowing access to 340 static HTML pages, 280 illustrations, hundreds of dynamic pages, and a few dozen Java and Javascript applications. Illustration 1 presents the three frames used. The right-hand frame contains a dynamic, collapsible **table of contents (index)** for the textbook portion of the system. (Hebrew is written right-to-left). The table of contents displayed, at first "load", the nine chapter headers. A click on a chapter header "unfolds" a list of links internal to that chapter. The bottom frame contains ten buttons pointing to various tools. The upper left hand frame is the main work area. Text pages, illustrations, and applications are displayed in this area . The system contained 1126 links. Of these, 233 links are available on the screen (the bottom ten pointers to tools, and 233 chapter headings in the table of contents).



Navigation Bar

Illustration 1: Screen snapshot of the opening page.

The textbook contained just under 30 "nodes" (or pages) on average for each of the 9 chapters. Each node consisted of 2 to 3 pages (screens) of text, accompanied by static or animated illustrations in the GIF format. Each node ends with a pair of buttons/pointers to the following and preceding "page" (node) such that a student may browse linearly. A Javascript element loaded with each page ensured that the display in the table of contents highlighted an updated indication of the title and context of the page displayed at any given time in the main area. Each chapter ended in a collection of links. **Review questions**, in a true/false format, were made available for each chapter. Each batch of about 20 questions allowed students to voluntarily and interactively complete their answers. A CGI script graded and provided feedback and the correct answers. Each access to the review questions generated a textual and numerical response to the student, and was recorded in a special log. A special **glossary** was built, to summarize definitions of the concepts covered in the course. The glossary contained over 100 terms, explained in both English and Hebrew. Each definition was accompanied by a link to the more complete discussion in context in the "book". A **search engine** allowed full text, simple and complex boolean search of the entire system. **Homework assignments** included required exercises in searching the Web, trying out and proposing improvements for interactive decision support systems, proposing a strategic information system, and designing DFD and ER graphics. The homework assignments were also used as an opportunity to require students to read and react to each other's work, by basing later assignments on other students' earlier reports. An **open discussion board** allowed a-synchronous communication among students and between instructor staff and students. The board served for the submission of all homework, and the publication of grades. An average student posted a dozen messages to this board during the semester, including textual messages, hypertextual links, and graphics, such as the DFD and ER designs. The interactive board served as the meeting ground where students read each others' homework, and were required to react to other people's work. Two more recent additions to the tools included in the learning environment are a map and a note-taking system. The **map** displays a visual chart of the chapters and pages in the textbook. Each page is color coded to represent frequency and date of previous visits. At the touch of the appropriate button on the bottom "navigating" frame a student may receive a summary of what they've read and when. The map provides a navigation aid that is both spatial (in the text context) and temporal. Furthermore, students may also view a similar map displaying class averages, providing an online, virtual equivalent of social comparison. Also on the map page are summaries of earlier uses of the review question sections, and the results (grades) reached in trying these questions. The **notes** subsystem allows students to mark-up the text. The purpose is to personalize and internalize the system's content. Analogous to the yellow highlighting so prevalent in college-reading, this system combines icon marking with textual (written, typed) notes, that are fully searchable. Both the notes and the mapping systems are individual and dynamic, owned by- and relating to a particular student at a specific point in time. These latter two systems were developed only recently, hence not included in the data reported here.

Methodology

In the following we report on usage and performance by three classes during the winter of 1996/7. These are undergraduate (BA), graduate (MBA) and executive MBA classes (see Table 1). Students' final grade in the class was used as the primary performance criterion. The final grade was based on performance on midterm and a final tests, and six homework assignments. Each of the components (tests and assignments) were graded by different people. The Pearson product-moment intercorrelations among components of the final grade ranged between 0.5 - 0.8. Usage statistics include access to pages and usage of the further interactive (CGI-based) applications, such as search mechanism and review questions. The access log records were purged of irrelevant records, such as those reflecting use by instruction staff, providing a clean database of "Requests". We calculated a listing of student access paths (visits), arranged chronologically. A further statistical manipulation allowed the creation of a Markovian matrix of transitions. In this matrix, each line represents an "originating page" in the text. Each column represents a "target page". Each cell in the matrix expresses a probability for the transitions from a given page to another. After collapsing for some irrelevant pages, the matrix contains 90000 cells, (the square of 300 pages). The values in each row sum to 1, the sum of probabilities of transfer from each page to all other pages.

A measure of reading nonlinearity (NL) was developed. Nonlinearity is defined as the transition ("navigation") to a nonadjacent node. Nonlinearity was calculated as the ratio of the sum of nonlinear transition probabilities to the total number of pages (300). A value of zero on this measure expresses perfect linearity. A maximal value of one stands for a user who never read to adjacent pages. Stated formally: with $A_{i,j}$ cells in the transition matrix of j rows and i columns, the nonlinearity measure (NL) is calculated as:

$$NL = \frac{\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i \\ j \neq i+1 \\ j \neq i-1}}^{m} (A_{i,j})}{\sum_{i=1}^{n} \sum_{j=1}^{m} (A_{i,j})}$$

Usage and performance correlations were assesed via Pearson product-moment correlations for the individual components, and by regressing the overall (final) grade on usage variables.

Results

Less than half of the undergraduates have used the internet prior to the course (see Table 2). Among MBA students 56% have used the internet before the course. However, in both groups about 70% reported that they will increase use of the internet in their work and study in the future. About one third of the students borrowed" usernames and passwords from classmates on occasion, a fact that sets upper limits to the reliability of access data. This phenomenon was significantly more prevalent among the MBA, graduate students. Home computing is widely diffused, with two thirds of the students reporting that they have a personal computer at home. Nevertheless, and despite the availability via the internet, more than half the accesses to the system were done from the fairly limited computer laboratories at school. A very large proportion of the students opted to create and use hardcopy versions of the text.

TABLE 2: Student self-report questionnaire summary

		BA*	MBA*
Number of questionnaires		81	66
Use only your own username?		72.*%	69.2%
Internet connection prior to course?	Yes	17.3%	33.3%
	Sometimes	23.5%	22.7%
	No	59.2%	44%
Use Internet in future?	Less	2.5%	3%
	No change	27.2%	27.3%
	More	70.3%	69.7%
Have PC at home?	Yes/No	63%/37%	66.7%/33.3 %
Primary reading menas	From screen	24.7%	18.7%
	Printed	25.3%	21.7%
	Photocopy	50%	59.6%

*Significantly different values in bold

In a questionnaire distributed at the end of the semester, we asked students in the BA and MBA classes for their evaluations of the system. Overall reactions were positive, even despite the numerous frustrating technical obstacles. The graduate students were slightly more appreciative of the system, though the differences were not statistically significant. When asked whether they would prefer to have such systems incorporated in the remaining portion of their studies, 70% of the MBA students, and 55% of the undergraduates responded positively. In open-ended questions the most frequent complaint was that this course demanded inordinate amounts of students' time.

The original access log contained almost 140,000 records. Of these we purged 9 erroneous records, 2686 records dated earlier than the beginning of each class, 2162 non-course related accesses, 6433 instructors' accesses, and 15820 failed password attempts. The remaining requests are distributed as displayed in Table 3.

TABLE 3: Access log records

	BA	MBA	EMBA
Total # of request, including graphics	21,700	14,487	10,573
Requests without graphics	15,434	10,114	7.736

Table 4 reports average and standard deviations of accesses per student, comparing the three classes.

TABLE 4: Summary of access log statistics

	BA*	MBA*	EMBA*
N (students) total in log	83	56	40
Average # page requests (per student)	170.9	269.25	152.79
std.	109.15	97.9	105.96
Review Questions:			
Average # page requests	13.26	11.53	15.66
std.	12.2	10.65	11.06
Grades on Review Questions:			
N (students)	66	41	26
Average grade (out of 100%)	63.6	58.3	67.96
std.	11.6	16.6	11.14
Search Engine use			
N (students	83	56	7
Average # requests	0.27	0.5	5.71
std.	0.78	1.11	7.13

*Significantly different values appear in bold

The three classes were similar in their use patterns. The only significant difference in use is in the amount of online searching within the system, which peaked for the EMBA class. The access records detailed in Table 3 were then used to create transition matrices. Each content file ("page in the textbook") is represented by a column and a row in the matrix. Each column represents an originating file, and each row represents a destination file. Each cell stands for the probability of transitions from the originating page (column) to the destination file (row).

A close examination of the resulting transition matrices indicates that most students tended to stick to the linear, traditional page-by-page order of reading material. The most heavily traveled path, the diagonal in these illustrations, was to read the textbook linearly, even though such use was not prescribed or even suggested. Clearly, too, the EMBA students were much more likely to diverge, and browse nonlinearly. The proportion of dots outside the main diagonal for this class is visibly larger.

The Non-Linearity measure (NL) displayed in Table 5 is a summative expression for the probability of moving between pages in a nonlinear fashion. A higher value on this measure indicates less linear reading, and more "purposive" browsing, using hypertextual links, searches, table of contents, glossary and index navigation. An analysis of variance test for the differences between the classes shows that the differences are significant. Post -hoc paired comparisons show that only the EMBA group differs significantly from the other two groups on reading the entire course.

TABLE 5: Non-Linearity averages per class*

NL(Standard div.)	**BA**	**MBA**	**EMBA**
The entire course	0.581 (0.101)	0.585 (0.099)	0.621 (0.088)
Only the textbook	0.570 (0.22)	0.593 (0.185)	0.622 (0.149)

*standard deviations in parentheses)

Table 6 indicates very high positive linear correlations among the use behaviors, and significant modestly positive linear correlations between the final grade and each of the usage behavior measures. This pattern replicates reliably across classes.

TABLE 6:Correlatios matrix between final grade and usage behavior

BA(N =84) MBA(N=56) EMBA (N=38)	Final Grade	Grades on revdiew questions	Requests for review questions	Requests discussion board	All requests
Grades on review questions	0.1953 0.2499 0.1103				
Requests for review	0.1611 0.0813 0.1229	0.6924 0.7267 0.7069			

questions					
Requests for discussion board	0.209 0.2286 0.3034	0.3487 0.4056 0.1708	0.389 0.293 0.3089		
All Requests	0.2461 0.2849 0.3428	0.4907 0.5602 0.395	0.4995 0.5042 0.5299	0.859 0.827 0.879	
Requests textbook only	0.2147 0.2672 0.2853	0.4123 0.4601 0.441	0.3345 0.4129 0.4886	0.545 0.490 0.472	0.888 0.887 0,826

Note: Significant correlations ($p<0.05$) are displayed in boldface. Significant correlations ($p<0.01$) are displayed in boldface and italics.

The final grade (paper-based tests and homework assignments) was regressed on a logarithmic transformation of the usage score (how many pages people read) and of the online review questions grade. Results of this regression are displayed in Table 7.

TABLE 7: Multiple Regression of Final Grade (dependent variable) on logarithmic transformation of usage and online review questions grades (independent variables)

	BA	MBA	EMBA	ALL
Multiple R	0.544	0.452	0.682	0.487
R Square	0.295	0.204	0.465	0.237
Adjusted R Square	0.270	0.155	0.383	0.222
F	11.561	4.124	5.66	16.48
Signif F	0.0001	0.255	0.171	0.000

Actual usage and average grade on the optional online review tests were entered into a regression model to predict the final grade. The independent variables were transformed, using the natural lograithmic function, to highlight the role of low and medium use and to compensate for the overstated effects of very heavy use or top grades. The regression model results are that the findings when the linear correlation was used are even more pronounced. The independent variables (use) predict upwards of 20% of the variance in the dependent variable, final grade.

Discussion

Online, web-based learning systems are feasible, and worked well here with three large classes of varying levels. Observations and informal reports of those participating in theese classes, as well as responses to written questionnaires all point to acceptance of the method and satisfaction with the tools. Of course employing online techniques for an MIS introductory class is desirable as an instance of the subject matter ("cobblers not going barefoot"). But the online tools enable more than just "doing as we preach". Our results indicate that at least some of the goals we set for ourselves are obtainable, that there is an acceptance of such tools among students, that the use of such tools engenders more openness to the use of other online tools, and that there is a correlation between amount of use and performance using the system and external learning criteria. A respectable proportion of the variance in an externally assessed final grade can be (at least) predicted by unobtrusive measures of online usage. Furthermore, a large portion of the students read the material offline, limiting in advance the variance in the predictor variables.

We have also validated earlier findings about limitations and need for further improvement. We learned that many students will still resort to offline reading, when given the option. We learned that creating an online environment is not just a matter of pouring formerly linear content into a hypertextual container. Nonlinearity is a goal that requires striving, and its level is an indication of the students' nature. Our students found that they devote more time to this course than they expected, and than is customary. This finding is related to predicted correlates of interactive systems (see e.g. Newhagen and Rafaeli, 1996). It remains unclear whether spending more time on a course is a positive or a negative outcome.

This evaluation study was unobtrusive. It made use of automatically accruing access logs as the main source of evaluation data. As such, our results are non-experimenetal. We have no controls built into the data collection phase, students were not assigned randomly to "treatments", and there are alternative possible explanations for the findings. This is the greatest limitation of this study. The direction of causality cannot be determined, and we must reserve our interpretations. It might very well be that the correlation between access log data and final grades is an indication of better students using the system, rather than of the system making better students. However, even this minimal interpretation of the results is favorable to the implementation of such environments, in the sense that this sort of an online system serves the needs of more advanced students. Furthermore, the fact that the cumulative average score on the voluntary review questions correlates with the final grade too, places more weight on the first causal interpretation of the correlations.

The three classes shared a common subject matter but differed in number, length and frequency of lectures, and the availability of in-lab tutorials. Mostly, though, the classes were distinguished by the nature of their students. The three classes can be rank ordered on the "experience" dimension. Younger undergraduates lack both the academic and the employment/real world experience their elder graduate student counterparts have. The Executive MBA students are even more mature. Consequently, undergraduates received more instruction hours and more individual tutoring. The three groups did not differ in how they used the system, with the exception of the search mechanism that was used more extensively by the EMBA group. The average student read about 160 pagesonline. However, the groups did differ significantly in the linearity revealed in their reading. The more mature EMBA students made more nonlinear use of the resources available. We tend to interpret nonlinear use as better, and would like to ascribe the difference to maturity (see also Hart, 1996 on this issue).

We found the use of asynchronous discussion groups to vary between the groups as well. The EMBA students made more use of the discussion system for in-group discussions of the study materials. Everett & Ahern (1994) suggested that experience with human to machine communication is an imported skill. This datum tends to support their observation. It also suggests that online systems such as intranets which students are likely to encounter in the real world will require adjustment, that such systems could be good practice, and that the implementation of online learning systems may be easier among more mature audiences, and may require more preparation for younger undergraduates.

Many of our findings need to be taken cautiously for another reason. The use of online learning systems is still a novelty. This results in caution on the part of students, and reflects growing pains of first-time-runs. Some of the system components are still under development. We had numerous system and communication crashes for the first months of the system life-cycle. Dealing with a non-English language on the Web imposes yet another layer of complexity on designers, maintainers, and students. Yet, novelty has its positive sides as well. In an environment of rapid change, there is no doubt we will see accelerating use of new technologies in learning. It is doable, and in the field of information systems, it is advisable too. Further work is needed on designing online implements to allow teachers to collaborate in teaching similar courses. Another area for future work are further applications of the availability of access logs generated during the use of online materials. For instance, we are currently working on the merging of navigation aids and access logs. We are interested in finding whether better, dynamically updated navigation aids enhance browsing, and will that, in turn, add to learning.

References

Acker, S.R. (1995). Space, collaboration, and the credible city: Academic work in the virtual university. Journal of Computer-Mediated Communication [On-line], 1 (1). Available: http://jcmc.huji.ac.il/vol1/issue1/acker/ACKTEXT.HTM

Alavi, M. (1994). Computer mediated collaborative learning: An empirical evaluation. MIS Quarterly, 18(2), 159-174.

Dee-Lucas, D. (1996). Effects of overview structure on study strategies and text representations for instructional hypertext. In J. F. Rouet, J. J. Levonen, A. Dillon & J. R. Spiro (Eds.),Hypertext and cognition. (pp. 73-108). Mahwah NJ: LEA Publishers.

Fletcher, J. D. (1996). Does this Stuff Work ? Some Findings from Applications of technology to Education and training. Proceedings of Conference on Teacher Education and the Use of Technology Based Learning Systems. Warrenton VA: Society for Applied Learning.

Kulik, A. J. (1994). Meta- Analytic Studies of finding on Computer Based Instruction. In E. L. Baker & H. F. O'Neil (eds.),Technology Assessment in Education and Training. Hillsdale, NJ: Lawrence, Erlbaum.

Musciano, C. (1996, March 1). Collecting and using server statics. SunWorld Online [Online], 10 paragraphs. Available: http://www.sun.com/sunworldonline/swol-03-1996/swol-03-webmaster.html [1997, March 25]

Newhagen, J.E., & Rafaeli, S. (1996). Why communication researchers should study the Internet: A dialogue. Journal of Communication, Vol. 46:1, pp. 3 - 15, also in Journal of Computer-Mediated Communication [On-line], 1 (4). Available: http://www.usc.edu/dept/annenberg/vol1/issue4/rafaeli.html

Novak, P. T. & Hoffman, L. D. (1996). New Metrics for New Media: Toward the Development of Web Measurement Standards, [Online]. Available: http://www2000.ogsm.vanderbilt.edu/novak/web.standards/webstand.html [1997, March 25].

Rouet, J. F. & Levanon, J. J. (1996). Studying and Learning With Hypertext: Empirical Studies and Their Implications. In J. F. Rouet, J. J. Levonen, A. Dillon & J. R. Spiro (Eds.),Hypertext and cognition. (pp. 9-25). Mahwah NJ: LEA Publishers.

Schutte, G. J.(1997). Virtual Teaching in Higher Education: the New Intellectual Superhighway or Just Another Traffic Jam? [Online]. Available: http://www.csun.edu/sociology/virtexp.htm [1997]

Shale, D. & Garrison, D. R. (1990). Education and communication. In D. Garrison & D. Shale (eds.), Education at a distance: From issues to practice (pp. 23-39). Malabar, FL: Robert E. Krieger.

Stehle, T. (1995). Getting Real About Usage Statistics, [Online]. Available: http://www.naa.org/marketscope/conaghan/stehle.html [1997, March 25].
Thomas, J. (1996), Introduction: A debate about the ethics of fair practices for collecting social science data in cyberspace, The Information Society, 12, 107-117.
Wheeler, B.C., Valacich, J.S., Alavi, M., & Vogel, D. (1995). A framework for technology-mediated inter-institutional telelearning relationships. Journal of Computer-Mediated Communication [On-line], 1 (1). Available: online. http://jcmc.huji.ac.il/].

Alternative Course Delivery: Teaching Courses By Internet - An Overview

By Bruce White and Tom Farrell, Dakota State University, Madison, SD

Abstract:

Over the past several years, many factors have come together to make course delivery over the Internet a reality. First the Internet (in the format of the world wide web and electronic mail) has made great advances; secondly, there have been improvements in communications; third, software for course delivery is also more readily available, and finally, the demand for non-traditional course delivery has increased.

In the Information Systems arena, many potential students already have jobs but want additional education in computing and related topics. The rapidity that defines change in information systems has many people interested in updating skills, but may not be able to attend classes in a traditional setting.

This paper presents some aspects of Internet course delivery from two instructors who have taught courses over the Internet. Topics to be presented include: course organization, course delivery, quality control, audience, marketing, and evaluation.

Distance Education:

There have been distance education courses for decades. The most primary form of distance education in the past 50 years has been the correspondence course. In this mode, the student receives a textbook, studies on his or her own time, and at appropriate times takes tests over the material. The tests, papers, and other assignments are mailed back to the instructor who grades the material and assigns a course grade. With television came the concept of televised courses, frequently offered through public stations television for college credit. Students watch the televised lecture, read the text and complete the assignments, tests and papers as in a correspondence course. This mode has been extended through cable television systems partially or wholly devoted to education, such as "Mind Extension University" - which offers courses through the combination of televised lecture and assigned course work. A further enhancement of the televised course has occurred with video recorders, where the student can tape the lecture material and review it at their convenience. Another option for distance education has been the "course in a box" concept - where students receive pre-packaged videotapes (or compact disks) in addition to other course material. The lecture material is on the videotapes and can be viewed by the student like a televised course. Some colleges have also delivered courses on a closed circuit or limited access television system, where students in remote sites can watch, and interactively participate in, lectures and discussions held on campus in an interactive television studio.

Internet Courses:

The latest change to distance education has been with the delivery of Internet courses. In this format, students receive lecture material through world wide web pages and electronic interaction with students. The major advantage of Internet distance education delivery as compared to correspondence courses or videotaped courses is that students can interact with instructors in a more timely fashion. While correspondence courses and televised courses had the option for students to mail in or call the instructor with questions, it was not always reasonable. The asynchronous communication of electronic mail means that students can ask questions as they arise from their reading or assignments and receive answers from the instructor in the next couple of days. A non-traditional student who would is working on an assignment late at night might not be able to contact the instructor during regular office hours, now has the ability to contact the instructor electronically during off-hours. Internet courses are delivered to the students home, office or wherever they have computer and Internet access. The burden of learning is placed on the student - to get the material, to study, to do assignments, and to interact with the professor.

According to a quick survey of the "Internet University" off of the world wide web, three are several institutions delivering courses by Internet. A listing of these institutions is given in Appendix A.

From this list in appendix At, we can see that Internet course delivery is a reality. Taking this to the next level, there are many courses offered over the Internet. Here is a brief list, with a more complete list given in Appendix B.

Internet Course Delivery - Examples:

At Dakota State Univesity, within the College of Business and Information Systems has already delivered five courses by Internet. These courses are CSC 150 Principles of Programming; CSC 221 COBOL I; CSC 222 COBOL II; INFS 368 Object Oriented Programming and CSC 475 Artificial Intelligence. The authors of this paper have been involved with the delivery four Internet courses: CSC 150 Principles of Programming, CSC 221 COBOL I, CSC 222 COBOL II, and a one credit CSC 270 Special Topics: Developing and Managing Your Web Site. In addition, <name deleted> University has offered other courses for academic credit including: ENGL 103 Composition; ENGL 303 Advanced Composition; MUS 100 Introduction to Music; and <more>. The reason <name deleted> University decided to offer courses by Internet was to fulfill the schools mission in Information Systems. The second reason was to meet the needs of non-traditional students in the region. As a public institution, the mission is interpreted as reaching to citizens in the state (and region), even though hindered from coming to campus by geography or by individual schedule.

Course Example #1: CSC 150 Principles Of Programming:

Principles of Programming is the first required programming course in both the information systems and computer science degree program at Dakota State University. Because of the mission of the institution, it also counts as a general education choice and is taken by business administration majors, health information systems majors and other students. The course involves development of computer logic and programming skills. Students develop approximately 10 programs using the C++ programming language, although the emphasis on developing techniques for application development and lessor on proficiency in C++.

As an Internet course, CSC 150 Principles of Programming uses a textbook (currently Programming in C++ in 12 Easy Lessons by Greg Perry), world wide web lecture and assignment pages (the URL is:http://courses.dsu.edu/csc150), and instructor interacts with students through electronic mail. The textbook comes with a C++ compiler for the programming assignments, and students will use their own word / text processing programs for developing algorithms and other analysis.

Principles of Programming has been offered four times, and is scheduled for subsequent offerings.

Course Example #2 - CSC 221 COBOL I:

COBOL I is the second required programming course in the Information Systems curriculum and COBOL II is the third - and has (as expected) the pre-requisite of COBOL I. COBOL I was offered for the first time as an Internet course for Spring semester 1997, and COBOL II was offered for the first time during summer semester 1997. Like the Principles of Programming course, the course uses a textbook (currently Comprehensive Structured COBOL by Horn and Gleason), world-wide web lecture and assignment pages, electronic mail for notes and instructor interaction. The textbook was selected with the student MicroFocus COBOL compiler so students could develop their applications at home. Course expectations include 14 COBOL programs, three tests (one written and two on-line tests), and on-line quizzes. The URL of the course is: http://www.dsu.edu/~whiteb/cobol/cobol.html.

CSC 221 COBOL I was offered for the first time during spring semester 1997 with an enrollment of 26 students, from eight states and one foreign country. Two students dropped the course, with 24 students completing the course. One student added the course late, but was still able to complete the class in the scheduled timeframe..

Expectations:

It is expected that the students read the textbook, access the Internet web pages, and interact with the Instructor as needed. In our current course organization, students are expected to follow the course timetable, but the course could be organized as an open-entrance, open-exit individualized course. The advantage of a course timetable is that students are expected to complete assignments in a reasonable fashion and take tests on time. The major disadvantage is that student's schedules sometimes do vary with work expectations and family commitments, plus the extra workload on the instructor's part to keep track of each student's progress. The expectations in the Internet courses that the authors taught was on a fairly structured time schedule, although there were some allowance for special circumstances. Tests were administered within a week time frame, and assignments had scheduled due-dates (and like traditional classes, some students failed to meet the scheduled due-dates).

Course Lecture Material:

As with many courses, the textbook is supplemented by lecture material with additional examples and illustrations. In a distance delivery class, this can be difficult to do. Straight correspondence courses assumes the student will acquire the desired material through the textbook and outside reading without additional lecture material. Televised courses supplemented the textbook with televised lectures. Internet courses supplement the textbook with world wide web pages and with interaction with students by electronic mail. In the author's experience, it takes as much (if not more) time to prepare Internet course web pages then it does to prepare classroom lectures. The author who teaches the Principles of Programming course frequently prepares lectures using PowerPoint presentations. He has transferred these presentations to the world wide web pages in a parallel fashion. He has used Microsoft's web authoring package FrontPage. The author who teaches the COBOL I and II courses in his on campus courses uses more program examples and traditional "board" lectures and presentations. He has incorporated these into web pages as well as sending program examples by electronic mail. He has used traditional HTML coding (on a Unix system), Microsoft FrontPage and Microsoft's Internet Assistant for Word. In both classes, numerous questions are submitted to the instructor by electronic mail and are answered within two days. The COBOL instructor also contacted most of his students by phone in the first week of the course to get to know them better and to adjust the course to their learning needs.

Student Evaluation and Grading:

In all of the courses the authors have taught by Internet, students are expected to complete assignments and take tests over the course material. The instructor of the Principles of Programming course has preferred "traditional" testing by using proctors to administer tests to students. He mails the tests to the proctors who verifies that the student taking the course is the proper student, and the proctors also monitor the student during the test as a precaution to cheating. The instructor of the COBOL Internet courses has one test that is handled through proctors, but uses an on-line test for the other two tests of the course. Students select the day and time to take the test, and the instructor sends the necessary components by electronic mail in time for the student to take the test. For example, in the COBOL I course for the first test, students selected a day and time within a week period that was most acceptable to their schedule. The instructor sent a partial COBOL program and data file to the student and the student had a two hour time period to finish the code and electronically send the materials back.

In comparing the performance of on-campus students and Internet students in the courses, the comparison seems very comparable. There are many factors that could introduce biases into a straight analysis. Most of the Internet students are non-traditional students and are highly motivated to complete the course for additional experience and or promotion. The on campus classes tend to be made up of traditional aged students and don't seem to have the same motivation. There are other factors as well, for example, many students in the COBOL I and II Internet classes have additional computer experiences including being employed as programming developers using other languages and desiring to learn yet another computer language, while students on campus come into the courses with limited experience.

Marketing The Courses:

While Dakota State University has a Director of Distance Education, much of the marketing for the courses was done independently of the distance education office. Because distance education by the Internet is a non-traditional situation, traditional marketing to older students is not as effective. For night classes aimed at the non-traditional working market, the university takes out ads in the area newspapers, plus sends flyers and advertising material to businesses and interested parties. For an Internet course, newspaper ads were considered of little value. Instead, the URL (Uniform Resource Location) of the course home web page was submitted to several search engines as a starting point for the marketing effort. Also students that had completed the first course (Principles of Programming) were contacted by electronic mail with information about the second course. A third method of advertising was to post basic information about the course in related usenet user newsgroups. For the first course, since it is a first and foundational course in Information Systems for most students at Dakota State University, flyers were sent to regional high school computer teachers. There were some students in the Principles of Programming course that were high school students wanting to get an early start on their collegiate coursework.

Quality Control:

In the Internet courses taught by the authors, they also taught the same courses on campus. Every attempt was made to make the course content similar. The tests and assignments were either the same or very similar between the Internet sections and the campus sections. In a distance education course of almost any nature, where the instructor does not see the student and interact with the student. the possibility of cheating exists. The written tests were monitored by proctors to verify that the person taking the test was the correct individual. That is, a professional COBOL or C++ programmer could not stop in at the proctor's office to take the test for the student. The programming assignments could well have been written by a person or persons other than the enrolled student, but the same situation could happen on campus as well. With most of the non-traditional students, they were taking the course to learn the subject matter, and to cheat in order to receive a higher grade would not have given them the skills they wanted.

Conclusion:

While a full analysis is hard to make, the early returns from teaching Internet courses from the experience at Dakota State University is that the courses indeed can be delivered, and delivered in a similar fashion to traditional courses. The authors find that while an Internet course will free an instructor from having regular class meetings, that a comparable amount of time (or even more) is spent developing web pages and communicating electronically with students. The authors feel that more courses will be delivered by the Internet in the future as students desire to gain additional education in a non-traditional setting.

There was additional time requirements on the instructors to read the many electronic mail messages from each student each week (with assignments, quizzes and course questions). There were occasionally technical questions that required additional assistance from the computing services staff, generally relating to lost electronic mail, or incorrect/incompatible file formats.

The authors will teach Internet courses again, but suggest to administrators that the workload factor for an Internet course be considered differently than a campus class, possibly giving double load credit the first time a course is taught by Internet. That is a three credit hour course would be considered six hours for workload on the first time it is taught, as the time commitment to create web pages and notes is significant. The second time the same course is taught, no additional workload is absolutely required. If the course is taught more than twice, then for the odd times it is taught, the workload should be doubled for the instructor generally changes notes, textbook, course approach or some other factor.

Class size might also be considered in addressing workload considerations. Internet classes have a significant overhead in sending and receiving electronic mail, and Internet classes over 25 students would have an almost overwhelming time commitment for the instructor interacting one-to-one with students.

Other Dakota State instructors have used listservs for class communication between students. The authors decided that for these courses that listservs might be counter productive for these courses and stayed with web pages and electronic mail. Classes where discussion and classroom interaction is important would be strengthened through using listservs. Questions that arose frequently were addressed through additional notes on the web page or through notes sent to the entire class through electronic mail.

All-in-all, teaching of Internet courses is an exciting new development for these authors, and it may be a foretaste of educational course delivery for the future.

Appendix A – Listing Of Institutions Offering Courses By Internet Delivery:

The following is a list of institutions offering courses by Internet. It would be reasonable to assume that additional institutions are now offering courses by Internet that are not listed here.

Antioch University
Brevard Community College
California Institute of Integral Studies
City University
Cal. State U-Dominguez Hills
Dakota State University
Edgewood College
Embry-Riddle Aeronautical University
University of Florida
Front Range Community College
Heriot-Watt University
Western Ill. University
University of Iowa
International. School. of Information. Management
International Society for Technology in Education
University of Massachusetts. - Dartmouth
New School. for Social Research
Norwich University
Nova Southeastern University
New York Institute of Technology
Penn. State University
University of Phoenix
Rochester Institute. of Technology
Rogers State College
Salve Regina University
Thomas Edison State College
University of Washington
Webster University
University of Wisconsin. - Madison
University of Wisconsin. – Stout

Appendix B – Listing of Internet Courses

The following is a list of computer related courses found on the Internet. It would be reasonable to assume that even more than this list are currently being offered.

Advanced AutoLISP Programming
Advanced PC Topics
Applied Database Management
Artificial Intelligence and Expert Systems
Assembly Language and Architecture
Authoring Systems and Curriculum Design
Authoring Systems Design
C Language Programming
C Programming
C++ Computer Programming
C++ Programming
C++ Programming Language
COBOL I programming
COBOL II programming
Computer Application of Learning Theory
Computer Concepts
Computer Concepts and Software Systems
Computer Graphics for Information Managers
Computer Information Systems
Computer Integrated Manufacturing
Computer Literacy and Educational Reform
Computer Management Information Systems
Computer Networks
Computer Security
Computer-Aided Software Engineering
Computer-Based Research and Statistics
Computer-Based Statistics
Computers and Information Processing
Computers and Problem Solving
Computers and Society
Computing Technology Facilities Planning
Courseware & Educational Programming
Data and File Structures
Data Center Management
Data Communication Technology
Database Searching Online
Database Systems
Database Systems Practicum
Developing a C Application
Distributed Database Management Systems
Essentials of Object-Oriented Technology
Human-Computer Interaction
Introduction to AutoLISP Programming
Introduction to C++
Introduction to Computer Conferencing
Introduction to Computer Programming
Introduction to Computers & Programming
Introduction to Programming
Introduction to Technology Systems
Introduction to WordPerfect; 6.0 for Windows
Learning HyperCard and HyperTalk
Learning Linkway/Linkway Live
Managing your own Web Site
Mathematics and Computing
Microcomputer Applications
Microcomputer Applications Processing
MIDI Composition: Creative Sequencing
Multimedia and Emerging Technologies
Network Management
Object Oriented Programming
Object-Oriented Applications for CIS
Office Automated Systems
Operating Systems
Programming Design & Validation
Programming Languages & Operating Systems
Software Sampler I
Software Systems Analysis & Design
Structured Programming in Pascal and Logo
Switching Technologies
System Test and Evaluation
Systems Analysis and Design
Systems Analysis for Educational Computing
Systems Design
Telephony (voice/data systems)
The UNIX Operating System: Introduction
Unix OnLine Course
Web Craft: Creating Your Own Web Pages

What is MIS?: Perspectives of Journal Editors

Uma G. Gupta
Jack and Joan McGraw Professor in IT
College of Business
Creighton University
2500 California Plaza
Omaha, NE 68178
(402) 280 2129
dcgupta@ecuvm.cis.ecu.edu

Eli Cohen
Visiting Professor of IS
Grand Valley State University
920 Eberhard Center
Grand Valley State University
301 W. Fulton
Grand Rapids MI 49504-6495
Eli_Cohen@acm.org

ABSTRACT

"What is information systems (IS)?" is arguably one of the most important questions that IS researchers, practitioners, and educators need to answer. In recent years, IS educators have often had to defend their teaching turf from potential intruders both within the business college and from without. The IS community as a whole is plagued with internal debates over issues such as the nature and scope of our field. Indeed, educators need to answer the question of whether IS education is a legitimate field of study. The relationship between MIS and other disciplines is both complex and fluid. This paper deals with the development of a methodology for identifying the reference disciplines of IS and the disciplines that in turn, are influenced by IS. The relationship between this paper and a forthcoming study of editors of IS journals is explained.

The Internal Debate: The Argument Against IS as a Discipline

Some would argue that IS is not a discipline at all. Dickson, Benbasat, and King write "Information systems, as an area of study, is clearly eclectic in nature and much material from other areas must be integrated together if it is to be successfully practiced and taught. They continue that when one traces the origin of the word "Management Information System" it does not go back to a technical term or concept, but rather to the word "Management".

King argues that information systems is probably not even a field, "but rather an intellectual convocation that arose from the confluence of interests among individuals from many fields who continue to pledge allegiance to those fields through useful ties of various kinds.

The appearance of the term "reference disciplines", in IS discourse, according to King, reflects that the area of MIS still lacks a "solid intellectual center" and the best way to address this issue is to "take the bull by the horn". But reference disciplines are critical for an evolving field for three reasons, as King explains. First, reference disciplines serve as an established source of intellectual capital; second, they provide the IS community with an "appeal to authority"; and finally, reference disciplines are an excellent way for identifying pockets of research that are uncharted. "Discipline is important for us, and obtaining it by reference is a perfectly sensible way for us to proceed, despite the inherently marginalizing consequence of our dependence on "outside" versus "inside" disciplinary traditions [King]. These are solid arguments that affect us as IS educators. AACSB no longer requires course work in IS, a point some deans view as excellent areas for reducing resources.

The Changing Discipline: IS as a Dynamic Entity

On the other hand, the field of MIS continues to evolve. Several environmental forces are driving this evolution. First, the technology itself is growing at a rapid pace, both in terms of the computing power of hardware and the enormous capabilities of software. Second, technological advances are compelling organizations to re-think how they do business, including organizational structures, management styles, and impact of technology on human behavior. Third, the rapid rate of the MIS evolution is forcing organizations to address the issue of how they can assimilate and benefit from technology [Sprague,].

The evolution of MIS has been not only rapid, but, more importantly, completely unplanned. Practitioners and researchers alike are struggling to address the critical issues that will shape the field in the coming years. Issues such as "What is MIS?" "What is the impact of MIS on other disciplines and vice versa?", and "What should we do as a community to strengthen the evolution of MIS?" call out for attention.

Even though the evolution of MIS has been rapid and unplanned, there are discernible trends, as shown in recent literature. Since the mid-1980s, the emphasis of MIS research efforts has shifted. For example, for the first time during the past two decades, the number of empirical articles exceeded the number of nonempirical articles essentially focusing on purely conceptual work. Empirical articles have been on the raise. However, there is a paucity of MIS-oriented theories is alarming because of the potential implications for quality experimental design, and the ultimate value of MIS research.

Current Thinking: The Debate Continues

As late as 1996, major articles debated the nature of IS. Yet, there tends to be consensus, at least among textbook authors, that MIS draws heavily on different source disciplines. These disciplines include computer science, management sciences, organizational behavior, behavioral science, management accounting, economics, and library science [Sprague]. Researchers argue that by drawing heavily from these disciplines, MIS will gain the respectability that most evolving fields must work hard to earn.

Is IS relevant? Relevance comes from successfully applying the findings of MIS to the real world. One of the characteristics of an evolving field is that research in such fields often lack rigor and this criticism continues to be levied against the field of MIS [Turner] Is it rigorous? Robey suggests that IS can best be understood as a colleague of its companion disciplines. Robey argues that there is a danger to looking inward in defining the field in isolation from other lines of inquiry. Such insulation works against its nature as a dynamic, evolving field of inquiry.

Some Research Into the Nature of IS

MIS is a relatively new field of research struggling with immaturity and ambiguity. Several researchers have examined the emerging field of MIS through citation analysis, research exploratory analysis. For example Culnan writes that MIS is in the pre-paradigmatic stage and lacks a consensus on the body of unique MIS knowledge [Culnan]. Other researchers have noted that MIS "has not made very significant progress as a scientific discipline."

Who Shall Judge: Editors as Arbiters

Our study focuses on perceptions of a sample of MIS researchers to address issues vital to the future of the field. What are the sub disciplines of MIS and our reference disciplines? Is there a consensus among the opinion leaders? Galliers writes that a good academic journal should disseminate new information directly to those who can apply it. "In an applied discipline, such as Information Systems, I would argue that it is important that we undertake research that is seen to be relevant by our colleagues in [IS Practice], as well as sufficiently scholarly by our colleagues in academia. This is the challenge associated with the term 'academic' in the field of Information Systems. while we wish to be scholarly, we do not wish to be labeled "unpractical," which is one of the meanings of the term 'academic.'"

We propose a study that asks editors of IS journals for their views on the similarity of various subdisciplines (such as information retrieval) to IS. These fields might include many that typically are not taught within a business school, such as web mastering, library science, and such. From this we wish to determine the psychological distance of such fields in the minds of these opinion leaders.

This problem could become circular. We need to know what is IS before we can determine which journals relate to IS. The solution we suggest is that we as the investigators do not make such a determination. Rather, the investigators will consider the population of all journals that have any of the following words in their titles: Information, Systems, Computer, or Data. Further, the investigators may wish to limit their population to editors of journals in the language English. To date, we have located 57 such journals.

Such an investigation would deal with perceptions of these influential gatekeepers. For many if not most professors, publication in IS journals is important for career advancement. Therefore the views of these individuals is of major concern.

We offer the following null hypotheses for such a study:
1. H0: The meaning and perception of information systems is uniform throughout the world.
2. H0: The definition of information systems and its role is the same regardless of the department affiliation (CS, IS, other) of information systems within the university environment
3. H0: The Barki et al. key word classification scheme captures all the variations of topics within IS.
4. H0: IS education is a significant issue in the eyes of editors.

At the conclusion of such a study, we will be able to provide a conceptual map of the field, with psychological distance measures. Such a map will help the reader better understand the interrelationships among IS, its reference disciplines, and its sub-disciplines.

References:

Dickson, G.W., Benbasat, I., and King, W.R., "The Management Information Systems Area: Problems, Challenges, and Opportunities," pp. 1- 8]

Galliers, R.D. (1994). "Relevance and Rigor in Information Systems. Research Community," in Business Process Reengineering - Information Systems Opportunities and Challenges: Proceedings of the IFIP TC8 Open Conference on BPR, Elseiver, Holland.

King, J.L. (1993) "Editorial Notes," Information Systems Research, 4(4), pp. 291-298.

Sprague, R.H. Jr., "Issues in Publishing for MIS Academicians", 203-205.

Turner, J.A., "Improving the Quality of Information Systems Research," pp. 91-97

Cohen, E. B. "IS as an Evolving Field". Proceedings of the SCI Conference
'97, Caracas, Venezuela, July 7-11, 1997.

An Information Systems Process Modeling Approach to Quality Classroom Instruction

Olu Omolayole
Department of Computer Information Systems
College of Business, Grambling State University
Grambling, Louisiana 71245.

Abstract

This paper presents a Quality Classroom Instruction (QCI) model. A combination of Instructional Design principles and Quality Control concepts with emphasis on service type organization is used to develop the model. Data Flow Diagram is used in designing the quality control and assessment process of the Classroom Instruction. A suggested list of variables that could be used to determine QCI and its feedback is presented.

<u>Key words and phrases:</u> assessment instrument, classroom ergonomics, classroom instruction, customers, data flow diagram (dfd), entity, factor, input, instructional delivery, output, primary stakeholder, process, qci feedback, quality, quality classroom instruction (qci), quality control (qc), questionnaire, reliability, secondary stakeholder, student, teacher and validity.

I. Introduction

The term "Quality" is defined in (ANSI/ASQC A3-1978, 1978) as "the totality of features and characteristics of a product or service that bear on its ability to satisfy given needs." The "service" that is under consideration is Classroom Instruction and the "given need" is for students to acquire appropriate knowledge and skills. The phrase "Classroom Instruction" is used to mean the instructional activities that occur within the four walls of a classroom, telecast studio, laboratory, gymnasium, auditorium, etc., involving teacher and students during a class-period with the purpose of teaching appropriate knowledge and skills to students.

The approach of this paper is to view Quality Classroom Instruction (QCI) as a service model that is based on the principles of Quality Control (QC) (ASQC, 1996; Feigenbaum, 1983; and Juran, 1988). QC by definition, is the "operational techniques and activities that sustain the quality of a service(in this case, Classroom Instruction) that will satisfy the given need(in this case, students acquiring knowledge and skills) " (ASQC, 1996). In industry, a QC program assures quality products/service for customers. The basic operating tenet of quality in a service type organization is that an organization cannot afford to dissatisfy its customers, otherwise it will be out of business.

An academic institution falls under the category of a service type organization. In general, the quality control component of a service type organization consists of a) the service provider(the organization), b)the service(the work performed by the service provider's representative), c) the customer and d) the benefit of the service to the customer (Juran, 1988; Omolayole, 1993 and Parasuraman, et al, 1985). The following equations are assumed in the model. The Service Provider or Organization = The Academic Institution. The Major Service Provided = Classroom Instruction. The customers = The Students. The Benefits of the Service = The Knowledge And Skills Acquired By Students. Students are the beneficiaries of the Classroom Instruction. They would, at the end of the service(Classroom Instruction), determine what is or what is not QCI. Students are sometimes referred to as Primary Stakeholder and the parents, institution, employers and others, who may have vested interest in the students' education, are called Secondary Stakeholder (Jenlink, 1995).

It has been argued by a few education professionals, through personal contacts, that students have influence on Classroom Instruction. Therefore, it would not be valid for them to determine the quality outcome of a Classroom Instruction. Of course, students do influence Classroom Instruction. It is a fact of life in the service industry. Customers generally have influence on the services provided them. For an example, a service may not go well if the customer is not cooperative with the service provider. The same example applies to Classroom Instruction. However, research (Bhada & Brightman, Fall 96 & Spring 97 and Marsh, 1984) has demonstrated that students can determine the quality of Classroom Instruction regardless of the influence they may have on the outcome of the instruction. The provision is that the instrument which the students would use to determine the quality of Classroom Instruction must have passed validity and reliability tests (Bhada & Brightman, Fall 96 & Spring 97 and Marsh, 1984). For an example, if the quality ratings of the Classroom Instruction were consistently high in courses which the students performed poorly, then the quality assessment instrument used is not valid.

A Data Flow Diagram (DFD), also known as an Information Systems Process modeling tool (Fertuck, 1992), is used to design the relationship among the classroom instructional factors for the QCI model. Each factor consists of variables that could be used to measure the quality outcome of a Classroom Instruction. A suggested list of the variables is provided in section IV. A formal definition of QCI is presented next.

II. The Definition Of QCI

Based on the reviewed literature (Godfrey, 1993; Omolayole, 1993; Parasuraman, et. al., 1985; Pyzdek, 1991; Riley, 1991; Shaughnessy & Sawyer, 1991; and Zeithaml, et. al., 1990), relating to the concept of quality in the service industry, QCI is defined in this paper as:

> *The degree to which the classroom instructional factors can present opportunity for students to acquire knowledge and skills of a course content, satisfactorily to the expectation of the students during the class period.*

The "classroom instructional factors" consist of the following.

(i) **E**; Classroom ergonomics and instructional resources.
E stands for the condition of the classroom environment where students are being taught, including the configuration of the classroom and the availability of teaching resources for the teacher to use.

(ii) **D**; Instructional delivery.
D concerns the relevance of the instructional delivery(including course objectives).

(iii) **A**; Class attendance.
A means that students must be in class during the class period.

The details of the QCI model are discussed next, using a DFD.

III. The QCI Model

Figure 1 shows the DFD of the QCI model. It consists of three entities and two processes. The following is the detailed description of the components of the DFD and the flow of the instructional factors that control QCI.

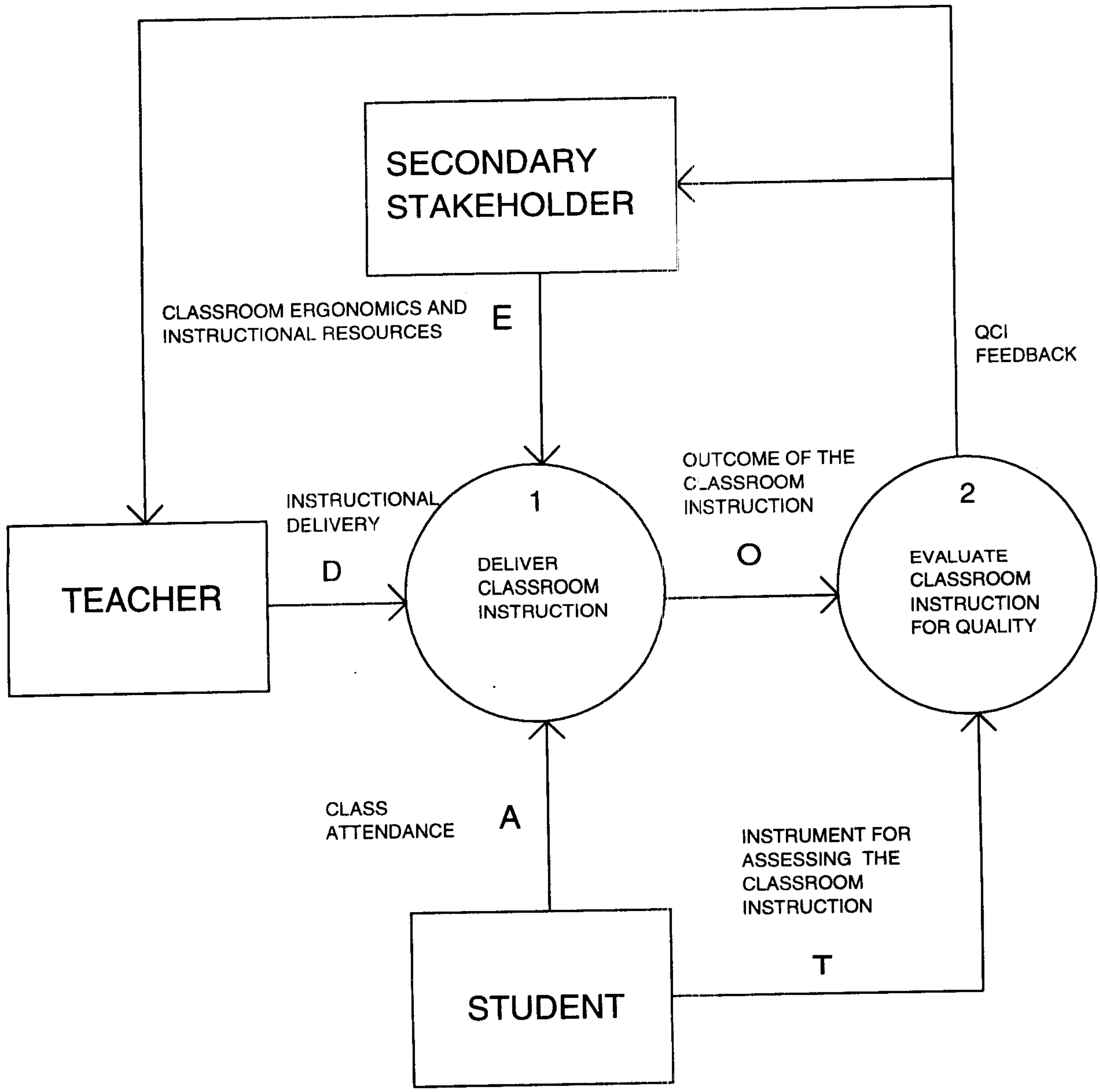


FIGURE 1 THE QUALITY CLASSROOM INSTRUCTION (QCI) MODEL

III.1 The Entities

The rectangular boxes in the figure identify the entities of QCI. They are SECONDARY STAKEHOLDER, TEACHER and STUDENT.

The SECONDARY STAKEHOLDER is the academic institution that has vested interest in the outcome of the Classroom Instruction. The institution provides the instructional resources and makes sure that the classroom environment is ergonomically fitted for instruction.

The TEACHER could be one teacher or a team of teachers that delivers instruction in the classroom. The teacher is expected to teach knowledge and skills of the course content to students, based on the instructional objectives developed for the course as documented in the course syllabus.

The STUDENT consists of students who are officially enrolled in the class. Some degree of class attendance is required for a student to participate in the quality assessment of the Classroom Instruction. The entity, STUDENT, represents customers in the model, while the SECONDARY STAKEHOLHER and the TEACHER represent the service provider.

III.2 The processes

The processes are identified by circles in the figure. Process 1 represents an instructional delivery subsystem. In its simplest form, it is the way the teacher teaches the course, supervises the students and manages the classroom activities during the class period in an ergonomically fitted classroom. Process 2 represents the evaluation of the output from process 1. The evaluation should occur at the end of the school term, using an assessment instrument.

III. 3 The data flow

The data flow in the figure consists of **E, D, A, O, T** and **QCI feedback**. As discussed in section II., **E, D** and **A** are the classroom instructional factors which are regarded as inputs to process 1. Specifically, **A** is a degree of class attendance over the school term that is required for a student to participate in the quality assessment of the Classroom Instruction. Each student taking part in the assessment should be officially registered for the course. Conceptually, a student who has a poor class attendance does not have any basis to take part in the evaluation of a Classroom Instruction which normally would include the evaluation of the teacher's classroom performance.

O is an output from process 1. It represents the cumulative outcome of the Classroom Instruction over the school term. Hence, the outcome is a reflection of the students'(the customers) classroom experience. It is the experience they would use to evaluate the Classroom Instruction for quality. Hence, **O** serves as an input to process 2.

T is the appropriate quality assessment tool that the students would use to evaluate the Classroom Instruction, at the end of the school term. Hence, **T** is also an input to process 2.

QCI feedback, which is the output from process 2, is the result of the students' evaluation of the Classroom Instruction for quality. The result would include some analysis(statistical and others) of what is adequate or inadequate about the Classroom Instruction. For an example, the statistical analysis of a result might show that the temperature of the classroom was unbearably too high throughout a school term and there was no air conditioner in the classroom. Such a feedback, should prompt the institution(SECONDARY HOLDER) to install an airconditioner in the classroom, to provide adequate temperature during future Classroom Instruction.

IV. QCI Variables

The QCI variables are a combination of the instructional delivery factor variables, the classroom ergonomics variables and some required degree of student's class attendance. An ad hoc list of the variables is presented in Table 1.

The instructional delivery variables were derived from parts of an Assessment Oriented Syllabus (Cunningham & Omolayole, 1997). Also included, are variables from the student questionnaire that many colleges and universities use for their end-of-the-school-term evaluation of courses. An Assessment Oriented Syllabus is viewed as an instructional tool that should specify the scope of knowledge and skills that a teacher is expected to teach in a course, including other instructional activities that would occur in the course. It should be useful in evaluating the learning outcome of the students.

Table 1: The QCI variables.

CLASSROOM ERGONOMICS AND INSTRUCTIONAL RESOURCES (E) Variables
The classroom is accessible to every student including the physically challenged.
The classroom ventilation, lighting and temperature are comfortable for lectures.
The classroom appearance is clean and well maintained for lectures to take place.
The classroom seats are comfortable and well arranged for lectures.
The class size(number of enrolled students) is not too large.
The instructional technology and/or teaching aids and resources are adequate for teaching.
The content of the syllabus is adequate and relevant to the course.
The required textbook is relevant to the course.
Instructor uses teaching aids for the lectures.
Instructor's appearance in class is appropriate.

INSTRUCTIONAL DELIVERY (D) Variables
The reason for offering the course is clearly stated.
The pre-requisite for the course is clearly stated.
The course objectives are clearly stated.
Lectures are relevant to the course objectives.
What is expected of a student in the course is clearly stated.
Assignments, case studies, projects and examinations are clearly explained.
Quizzes and examinations are relevant to the materials already taught in the course.
Grades are assigned fairly and impartially.
Instructor displays a clear understanding of the course.
Instructor has an effective style of presentation.
Instructor speaks audibly and clearly.
Instructor displays enthusiasm when teaching.
Instructor makes good use of examples and illustrations.
Instructor discuses graded material quickly enough for students to benefit from it.
Instructor deals fairly and impartially with students.
Instructor has respect for individual students and encourages discussion of different views.
Instructor identifies major or important points in the course.
Instructor motivates students to do their best work.
Instructor explains material clearly.
Class lectures contain current information, including recent developments in the area.
Students are free to ask questions and encouraged to contribute to learning in the class.
Instructor allows time for questions, answers and discussion.
Course assignments are interesting and stimulating.
The course builds an understanding of concepts and principles.
The instructor does not come late to class.
The instructor has the credentials and qualification to teach the course.

CLASS ATTENDANCE (A) Variable
A degree of class attendance is necessary for a student to receive enough teaching to pass the course.

The classroom ergonomics variables are those elements that exist in the classroom environment. The elements could be engineered to facilitate learning (Jillson, 1993 and Knirk, 1992).

V. Conclusion

The idea of QCI, presented in this paper, is in its experimental stage. It uses student-centered quality control principles and Data Flow Diagram to model QCI. The design of the model is unconventional, compared with the traditional education concept's approach to QCI design. It is expected that the approach, discussed here, would lend itself to an easier method of assessing QCI.

The next task is to conduct a student survey to determine the final QCI variables from those listed in Table 1 that students would consider to be important for evaluating QCI. A QCI questionnaire will then be developed using the important variables. College professors that are considered to be distinguished teachers would validate the questionnaire. Then experiments will be performed to test the reliability and effectiveness of the model.

VI. References

ANSI/ASQC A3-1978. (1978). Quality Systems Terminology. American Society for Quality Control, Milwaukee, Wisconsin 53202.

ASQC Statistics Division. (1996). Glossary and Tables for Statistical Quality Control, 3rd edition. ASQC Quality press, Milwaukee, Wisconsin 53202.

Bhada, Y. & Brightman, H. J. (Fall 96 & Spring 97) "College of Business Teaching Workshop." Grambling State University, Grambling, Louisiana 71245.

Cunningham, B., & Omolayole, O. (1997). "An assessment oriented syllabus model for business courses." Submitted to the Journal of Education for Business. Washington, DC 20036-1802.

Feigenbaum, A. V. (1983). Total Quality Control, 3rd edition. McGraw-Hill, New York.

Fertuck, L. (1992). Systems analysis and design: with CASE tools. Wm. C. Brown Publishers, Dubuque, Iowa.

Godfrey, A. B. (1993). "Customer satisfaction, customer retention, or customer loyalty?" Proceedings of the 1993 Trustee Series, edited by the conference board, New York: The conference board, February.

Jenlink, P. (1995). Systemic Change: Touchstones for the future school. Skylight Training and Publishing, Inc., Palatine, Illinois.

Jillson, J. (1993). "Instructional ergonomics." Business Education Forum. v47, (3), pp34-35, February.

Juran, J. M. (1988). Quality Control Handbook, 4th edition. McGraw-Hill, New York.

Knirk, F. G. (1992). "Facility requirements for integrated learning systems." Educational Technology, v32 (9), pp26-32, September.

March, H. W. (1984). "Students' Evaluations of University Teaching: Dimensionality, Reliability, Validity, Potential Biases and Utility." Journal of Educational Psychology, v76, pp707-754.

Omolayole, J. O. (1993). "On the definition of Software Quality: From the end-user's perspective," Proceedings of the 1993 Midwest Computer Conference. Normal, Illinois.

Parasuraman, A., Zeithaml, V. A., & Berry, L. L. (1985). "A conceptual model of service quality and its implications for future research." Journal of Marketing. Fall, pp. 41-50.

Pyzdek, T. (1991). What every manager should know bout quality. ASQC quality Press, Milwaukee, Wisconsin 53202.

Riley, R. G. (1991). "An exploratory study of quality assurance methodology in therapeutic recreation using the Delphi technique." UMI dissertation service. Ann Arbor, Michigan.

Shaughnessy, E. P., & Sawyer, D. A. (1991). "Operational Profile: The key to customer-oriented quality assessment." SQT, Pre-print.

Zeithaml, V. A., Parasuraman, A., & Berry, L. L. (1990). "Delivering quality service: Balancing customer perceptions and expectations." New York: Free Press.

The Tasks of Information Systems Professionals in Malaysia

Graham K. Winley, Sim Kim Lau, Ang Y. Ang
Department of Business Systems, University of Wollongong
Wollongong NSW 2522, Australia

Abstract

This paper presents the results of an empirical study into the present and future tasks expected of information systems professionals in a range of Malaysian organizations. Information was collected using a three round Delphi study technique across a period of one year. The views of senior executives working in general management positions as well as senior personnel working in information systems positions are analyzed and compared. The results are related to the present and expected future profiles of these organizations and are of relevance to those responsible for IS curricula in academic institutions as well as commercial providers of education and training courses in this important South-east Asian nation.

Introduction

A considerable body of literature has developed, particularly in the past ten years, which examines the nature of the Information Systems (IS) discipline, IS curricula and the education and training needs of IS professionals in an environment which is dynamic in terms of technological innovations and changing organizational structures. The majority of published studies relating to these issues have been conducted in a range of developed nations [6, 7, 10, 11, 12, 13, 14, 15]. However published studies conducted in developing nations are less numerous [1, 2, 3, 4, 5, 8, 17].

Many studies have focused on the design of IS curricula and have been influenced by information gathered from IS academics and practicing IS professionals. These studies have identified broad areas of skills and knowledge often categorized as systems development, systems management, technical and user support. Some of the major issues identified include: a perceived gap between IS curricula objectives and the needs of practicing professionals [16]; the need for graduates to have high level communication skills [9]; the current balance between the emphasis placed on these four categories of skills, especially the balance in curricula between systems management and technical and user support skills [17] and the need to provide problem-based real-life experience project in IS courses.

The study presented here examines the tasks expected at present and in the near future of IS professionals in a range of Malaysian organizations. This Delphi study, involving three rounds, was conducted throughout 1994 with senior personnel from the general management (GM) and IS management areas in order to obtain a broad view of the role of the IS professional in these organizations and to enable comparisons of the views expressed by these two groups of professionals. The results of this study are intended to assist in understanding the present and future requirements of the IS professional in Malaysia and are of particular relevance to those concerned with IS education and training.

Methods and Analysis

Initially forty-five tasks were identified as being representative of the range of duties commonly performed by IS professionals. These tasks were developed from the most important tasks identified in the extensive Australian study [6] and were classified into four categories defined as Systems Development (SD), Systems Management (SM), Technical Support (TS) and User Support (US).

Senior personnel from the general management (GM) and IS areas in a range of Malaysian organizations (manufacturing, engineering, banking and finance, wholesale and retail, transport and public service) were asked to rate each task on a scale from 1 (unimportant) to 4 (very important) in terms of the duties expected of IS professionals in their organizations at present and in five years time. In addition they were invited in the first round of this three round Delphi study to add tasks that should be included or delete tasks which should not be included.

Representatives from the GM and IS areas were included in order to obtain a broad organizational view of the roles of IS professionals and not just IS managers views of their own area. The three round Delphi study was used to allow participants to reconsider their views in the light of earlier responses and this approach also encouraged participants to be involved and more attentive to the study.

Responses to round one resulted in five additional tasks being added to the list and the subsequent two rounds provided an opportunity for respondents to rate the fifty tasks with full knowledge of all responses received in rounds one and two respectively. The results presented here represent the responses at the conclusion of the third round. For each group and for each task a present and future mean rating was calculated together with the associated standard deviations. A Wilcoxon signed rank test was used to test for significant differences between the present and future ratings of each task assigned by the GM group and the same procedure was applied to the corresponding ratings provided by the IS group. t-tests were used to test for significant differences between the mean ratings of each task assigned at present by the GM and IS groups and the same procedure was used to test for significant differences between the future mean ratings of each task assigned by the two groups.

Using the mean ratings each task was ranked from 1 (most important) to 50 (least important), Spearman rank order correlation coefficients were used to compare the present and future rankings of each group as well as to make comparisons between the two groups on these rankings.

Results and Discussion

There were fifty one respondents in the GM group and thirty seven in the IS group. The respondents come from forty-four different organizations. The majority of respondents were from organizations designated as manufacturing/engineering, banking/finance with others representing wholesale and retail, transport and public service organizations. In each case participants represented organizations with a very large number of employees. These organizations used systems based on mainframes, minicomputers and microcomputers with a strong emphasis on minicomputers and microcomputers at present and an increased emphasis on microcomputers expected in the future. Local and wide area networking were dominant features at present and were expected to increase in the near future.

The majority of organizations indicated that they developed their own applications as well as purchasing systems from vendors. They followed standard systems development methodologies when developing their own systems and made extensive use of project management and CASE tools as well as 4GL and application generators. They indicated an increased high use of project management tools, 4GL and application generators in the future and a less important role for CASE tools. COBOL and RPG were the most common languages used with a high and uniform usage being made of word processing, spreadsheets and database software packages.

Tables 1(a) to (d) indicate the fifty categorized tasks together with the present and future mean ratings and rankings assigned to each task by the GM and IS groups. For simplicity standard deviations have been omitted from these tables noting that these were all less than 1 which indicated a quite reasonable level of agreement on the task ratings. The probabilities associated with the Wilcoxon tests have been omitted and instead an asterisk on a future mean rating is used to indicate a significant change in mean rating from present to future at a significance level of 0.01.

The results of the t-tests used to compare the GM group's present (future) mean ratings with the IS group's present (future) mean ratings on each task are not shown for each task since for most tasks the differences between the mean ratings assigned by the two groups were not significant at a 0.01 level of significance. The exceptions were: the present rating of the SD task determine client/user needs, which was rated significantly higher by the GM group than the IS group; the future rating of the SD task, define data requirements, which was rated higher by the IS group than the GM group, and the future rating of the SM task identify information strategy again rated higher by the IS group than the GM group.

Rankings, derived from the mean ratings, have been adjusted for tied ranks and an asterisk is used on a future rank if the task has changed its rank by 10 or more positions from present to future.

Mean Ratings of Tasks

The mean rating of all tasks in the GM group, with the exception of maintain existing programs, have increased from present to future. The mean ratings of all tasks in the IS group (except maintain existing programs and code and compile programs) have also increased. The mean ratings in the GM group did not show any significant increase from present to future. This is in contrast to the IS group which shows that the mean ratings of 50% of the tasks has increased significantly from present to the future (see Table 1a).

None of the mean ratings of the seventeen SM tasks have decreased in either group. Although the mean ratings in the GM group have increased, a significant increase occurred only for the task plan and manage project teams. On the other hand the IS group shows significant increase in 82% of the mean ratings from present to future for SM tasks (see Table 1b).

From Table 1(c) we see that neither group shows any significant decrease in the mean rating of TS task. Seventy-three percent of TS tasks in the IS group show significant increase in mean rating from present to the future. The GM group only show significant increase in the mean rating of the single task keep abreast of new technologies.

The US tasks (see Table 1d) in both groups did not show any significant decrease in mean ratings from present to future. The mean ratings of tasks relating to analyzing and preparing for training needs together with identifying the effects of systems on other business units within the organization have significantly increased from present to future in the IS group. The mean ratings from present to future in the GM group did not show any significant increase.

From Tables 1(a) to (d), we see that the total number of important tasks (mean rating of 3 or more) across all categories doubled from the present (21) to the future (42). The GM group identified 17 important tasks at present and increased this to 39 in the future with a corresponding increase from 11 to 38 for the IS group.

Tasks	GENERAL MANAGEMENT				IS MANAGEMENT			
	PRESENT		FUTURE		PRESENT		FUTURE	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
Determine client/user needs	3.50	1	3.57	3.5	2.89	15	*3.51	7
Document information needs	3.07	10	3.43	8	3.00	11	*3.43	9
Identify problems in the existing system	3.07	10	3.21	*25.5	3.11	5	3.24	*20
Identify user functional requirements	2.93	20	3.21	25.5	3.03	10	3.25	18.5
Write functional specifications	2.79	32	3.07	34	2.89	15	3.11	*33
Prototype systems, including screens & reports	2.86	26	3.07	34	2.57	46	*3.11	*33
Define data requirements	2.86	26	2.93	*40.5	2.86	18	*3.33	13
Design the data model	2.71	36.5	2.86	44	2.69	34	*3.25	*18.5
Design database	3.07	10	3.29	19.5	3.08	6.5	*3.46	8
Review structure of database	2.79	32	3.29	*19.5	2.51	47	*3.14	*30.5
Write program specifications	2.86	26	3.00	*37.5	2.70	31	2.95	*41.5
Code and compile program	2.71	36.5	2.86	44	2.84	19.5	2.84	*46
Write program testing plan for entire system or individual programs	2.79	32	3.14	29.5	2.70	31	*3.00	37
Develop test procedures	2.86	26	3.29	19.5	2.59	43.5	2.95	41.5
Prepare and run test data	2.93	20	3.29	19.5	2.70	31	2.97	39.5
Maintain existing programs	3.00	15.5	2.71	*48	2.97	12	2.97	*39.5

Table 1(a): Systems Development Tasks

Task	GENERAL MANAGEMENT				IS MANAGEMENT			
	PRESENT		FUTURE		PRESENT		FUTURE	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
Identify information strategy	3.21	2.5	3.43	8	3.05	8.5	*3.78	1
Identify key support systems for business environment	3.14	5	3.71	2	3.24	1	*3.70	3
Identify & prepare overall technical architecture plan	3.14	5	3.21	*25.5	2.78	24	*3.22	22
Define human, equipment or machine resource requirements	2.86	26	3.36	*13.5	2.95	13	*3.30	15.5
Provide advice to management	2.93	20	3.43	*8	3.16	2.5	*3.59	4.5
Write a feasibility study	2.79	32	2.93	40.5	2.59	43.5	*2.92	43.5
Write cost/benefit analysis	2.57	44	3.08	*32	2.62	40	*3.14	*30.5
Establish procedures to restore the database after system failure	3.21	2.5	3.50	5	3.16	2.5	*3.73	2
Plan a preventative maintenance program	3.07	10	3.36	13.5	2.89	15	*3.41	11
Plan for development, implementation and installation phases	3.07	10	3.29	19.5	2.86	17	3.22	22
Manage network	3.07	10	3.43	8	2.76	26	*3.41	*11
Negotiate contracts with vendors	2.71	36.5	3.07	34	2.78	24	3.05	*35
Plan and manage project teams	2.86	26	*3.57	*3.5	3.08	6.5	*3.57	6
Audit computer systems	2.64	40	3.14	*29.5	2.65	37.5	*3.19	*26
Present reports & demonstrations	2.36	49.5	2.86	44	2.59	43.5	*3.00	37
Plan cost reduction programs	2.64	40	3.36	*13.5	2.68	35.5	*3.30	*15.5
Write a request for proposal from vendors & suppliers	2.57	44	2.79	47	2.65	37.5	2.86	45

Table 1(b): Systems Management Tasks

Tasks	GENERAL MANAGEMENT				IS MANAGEMENT			
	PRESENT		FUTURE		PRESENT		FUTURE	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
Act as a technical consultant for internal or external groups	2.64	40	3.14	*29.5	2.59	43.5	*3.19	*26
Identify and select hardware alternatives	3.00	15.5	3.43	8	2.75	27	*3.14	29
Identify system software alternatives	3.00	15.5	3.36	13.5	2.70	31	*3.19	26
Identify and select software application alternatives	3.07	10	3.38	11	2.81	22	*3.27	17
Maintain existing equipment	2.79	32	2.64	*49.5	2.84	19.5	2.92	*43.5
Design wide area network	2.57	44	3.00	37.5	2.68	35.5	*3.19	26
Design local area network	3.00	15.5	3.21	*25.5	2.78	24	*3.41	*11
Maintain network	2.93	20	3.29	19.5	2.73	28	*3.32	*14
Quote the cost of an item of equipment	2.57	44	2.64	49.5	2.64	39	2.67	*50
Write technical reports	2.50	47	3.00	*37.5	2.41	50	2.76	48
Keep abreast on new technologies	3.05	5	*3.86	1	3.14	4	*3.59	4.5

Table 1(c): Technical Support Tasks

Tasks	GENERAL MANAGEMENT				IS MANAGEMENT			
	PRESENT		FUTURE		PRESENT		FUTURE	
	Mean	Rank	Mean	Rank	Mean	Rank	Mean	Rank
Identify other business units affected by project	2.57	44	3.14	*29.5	2.61	41	*3.00	37
Prepare and/or maintain user documentation	2.86	26	3.00	*37.5	2.81	21	3.11	*33
Assist users	2.93	20	3.29	19.5	3.05	8.5	3.22	*22
Analyze training needs	2.71	36.5	3.29	*19.5	2.70	31	*3.19	26
Prepare and maintain material for training	2.43	48	2.86	44	2.43	49	*2.73	49
Conduct training sessions	2.36	49.5	2.86	44	2.49	48	2.81	47

Table 1(d): User Support Tasks

Table 2 shows the number of tasks in each category which have been identified by the GM and IS groups as important at present or in the future. At present there is only very moderate agreement between the two groups about important tasks.

The groups agreed that the three SD tasks relating to the identification of problems in existing systems, the documentation of information needs and the design of the database are important. They also agreed on three important SM tasks at present which address the identification of an information strategy and key business support systems together with the establishment of procedures to restore the database after systems failure. Only one TS task was considered important by both groups (keep abreast of new technologies) and although the GM group identified four other TS tasks as important the IS group did not rate any others as important. The additional four important TS tasks identified by the GM group related to the identification and selection of hardware, systems and application software and the design of local area networks. The GM group did not rate any US tasks as important at present and the IS group rated assist users as the only important task in this category.

The groups showed much closer agreement on their future ratings of important tasks across all task categories than they did on their present ratings. All tasks considered important at present continued to be rated as important in the future with the exception of maintain existing programs which was considered important at present by the GM group but is not considered important by either group in the future. Only the GM group considered the three SD tasks write program specifications, develop test procedures and prepare and run test data as well as the TS task write technical reports as important in the future. On the other hand only the IS group considered the two SD tasks define data requirements and design the data model together with the single SM task present reports and demonstrations as important in the future.

	Task category	Present	Future
Included by GM but not IS	Systems Development	2	3
	Systems Management	4	0
	Technical Support	4	1
	User Support	0	0
Included by IS but not GM	Systems Development	1	2
	Systems Management	2	1
	Technical Support	0	0
	User Support	1	0
Included by both groups	Systems Development	3	9
	Systems Management	3	14
	Technical Support	1	8
	User Support	0	4

Table 2: Number of Important Tasks

Rankings of Tasks

The results so far have focused on important similarities and differences between and within the two groups based entirely on the mean ratings assigned to tasks by the groups. However we have not directly considered the relative importance of tasks in relation to other tasks and this analysis is based on using the mean ratings to rank the tasks in order of importance (1 most important to 50 least important). We emphasize the need to consider the relative importance of tasks by noting that although a task mean may have increased significantly this tells us nothing about any variation that may or may not have occurred to the relative importance of the task in relation to other tasks.

The present and future untied rankings of all tasks are included in Tables 1(a) to (d) and using these rankings Table 3 presents the values of correlation coefficients. From Table 3 we see that at present and in the future the groups are in close agreement on the relative importance of the fifty tasks. The present and future ranked lists for the GM group are in close agreement and that the same is true for the IS group.

GM vs. IS Management		Present vs. Future Rankings	
Present rankings	Future rankings	General Management	IS Management
0.75 *	0.42 *	0.71 *	0.46 *

Table 3: Correlation Coefficients

* significant at 0.01 level

Consequently, although the mean ratings of tasks have changed, and in some cases significantly, both groups are in agreement about the relative importance of all the fifty tasks at present and in the future and neither group has significantly changed the relative importance of tasks across that time span.

Highly ranked tasks

Highly ranked tasks are those ranked in positions 1 through 15. We note that about 40% of all tasks are ranked highly at present and in the future by both groups and with few exceptions all tasks ranked highly are also considered to be important (mean rating of 3 or more). The only exceptions are the three SD tasks determine client/user needs, write functional specifications and maintain existing programs together with the two SM tasks define human, equipment or machine resource requirements, plan a preventative maintenance program all of which are highly ranked by the IS group at present but are not considered to be important at present by that group.

At present the majority of tasks considered to be important are also highly ranked. The few exceptions being tasks related to identification and selection of hardware and systems software alternatives, the design of local area networks and the maintenance of existing programs all of which are presently considered to be important but only just fail at position 15.5 to be highly ranked by the GM group. In the future for both groups important tasks are not necessarily highly ranked.

Table 4 indicates the actual number of tasks in each category which have been highly ranked by both groups or by one group only. From Table 4, we see that at present they agreed on the four highly ranked SD tasks determine client/user needs, document information needs, identify problems in the existing system and design database and they agreed that the first two of these tasks are highly ranked in the future. At present the IS group also highly ranked the tasks identify user functional requirements, write functional specifications and maintain existing programs and in the future unlike the GM group they continued to emphasize design database and included define data requirements as a highly ranked task.

The level of agreement between the groups on highly ranked SD tasks decreased overall. The reverse occurred for highly ranked SM tasks. At present the groups agreed that the four tasks identify information strategy, identify key support systems for business environment, establish procedures to restore the database after system failure and plan a preventative maintenance program are highly ranked SM tasks and they also agreed to continue to rank these tasks highly in the future along with the additional three tasks manage network, provide advice to management and plan and manage project teams. The first of these three tasks was highly ranked by the GM group at present and the other two were highly ranked by the IS group at present. Both groups agreed at present and in the future that the single TS task keep abreast of new technologies is highly ranked. At present the GM group included the additional task identify and select software application alternatives and in the future they again emphasized this task as well as the tasks identify and select hardware alternatives and identify and select systems hardware alternatives and identify and select systems software alternatives. The IS group added design local area networks and maintain networks as future highly ranked tasks. Overall the level of agreement between the groups was maintained with respect to highly ranked TS tasks. The responses in the US category were quite surprising. In the future neither group has highly ranked any of these tasks and at present only the single task assist users was highly ranked and even then only by the IS group.

	Task category	Present	Future
Included by GM but not IS	Systems Development	0	0
	Systems Management	3	2
	Technical Support	1	3
	User Support	0	0
Included by IS but not GM	Systems Development	3	2
	Systems Management	3	0
	Technical Support	0	2
	User Support	1	0
Included by both groups	Systems Development	4	2
	Systems Management	4	7
	Technical Support	1	1
	User Support	0	0

Table 4: Number of Highly Ranked Tasks

From Table 5 we see that both groups have more upward compared to downward movement across the SM and TS categories with the reverse occurring for both groups in the SD category where significant movement is predominantly downward. They differ with respect to the US category where upward movement predominates for the GM group and the IS group has only made significant movements in the downward direction.

We now consider those tasks which have changed their rankings by at least 10 positions either upward, which indicates an increase in relative importance, or downward, which indicates a decrease in relative importance. The number of such tasks by category for both groups is shown in Table 5.

		Task Category				Totals
	Movement	SD	SM	TS	US	
General Management	Up	1	6	2	2	11
	Down	4	1	2	1	8
IS Management	Up	3	4	3	0	10
	Down	5	1	2	2	10

Table 5: Significant Changes in Task Rankings

From a closer examination of the tasks significantly moved by the groups we see that both groups agreed that the single SD task review structure of database move upward while the IS group moved upward the two additional tasks design data model and prototype systems, including screens and reports. They agreed that the three tasks identify problems in the existing system, write program specifications and maintain existing programs move downward with the GM group also moving downward the task define data requirements and the IS group moving downward write functional specifications and code and compile programs. In the SM category the GM group moved define human, equipment or machine resource requirements, provide advice to management, plan and manage project teams, write cost/benefits analysis, audit computer systems and plan cost reduction programs upward. The IS group agreed with the upward movement of the last three of these tasks but also moved manage network upward. Each group moved a single task downward in the SM category. For the IS group it was negotiate contracts with vendors and for the GM group the task was identify and prepare overall technical architecture plan. In the TS category they agreed to move acts as a technical consultant for internal or external groups upward and maintain existing equipment downward. In addition the IS group moved maintain network upward and quote the cost of an item of equipment downward. The GM group moved the additional tasks write technical reports upward and design local area networks downward which is in sharp contrast to the IS group who moved design local area networks upward. In the US category both groups moved prepare and/or maintain user documentation downward with IS group also moving assist users in the downward direction. There were no US tasks moved significantly upward by the IS group while the GM group moved identify other business units affected by the project and analyze training needs upward.

Conclusion

Our results show that at present the GM group placed more emphasis on systems management and technical support tasks than the IS group while the IS group placed more emphasis on systems development and user support tasks. However, both groups placed less emphasis on technical support and especially user support tasks than might be expected. For the future the GM group placed more emphasis on systems management tasks than the IS group while the IS group placed more emphasis on systems development tasks. Comparing the responses for the present with those for the future we see that both groups increased emphasis on systems management and technical support tasks. They both decreased emphasis on systems development tasks.

In conclusion, systems management and development tasks are perceived to be important by both IS and general management professionals at present. However, non-IS professionals (general management group) perceived that systems management and technical support tasks will be more important in the future. On the other hand, IS professionals continue to place more emphasis on systems management and systems development tasks in the future. This confirms the need to place more emphasis on systems management issues. A large number of organizations in Malaysia are currently experiencing extensive growth in systems development and implementation of IT applications in all levels of operations. This is reflected in the perceptions of both IS and general management on the importance of systems development tasks. Relating these trends to information collected on the present and future profiles of organizations participated in the study, we can see the relationship between the increase need for systems management and technical support skills and the increase use of project management tools and local and wide area networking. The relative lack of emphasis on user support skills is surprising however it may be explained by a growth in the number of competent end users in these organizations. Systems development skills, especially those relating to programming tasks, are changing possibly due to the increased use of 4GL and application generators, the use of CASE tools and the reduced need to develop systems entirely within the organization.

Finally what can this study contribute to the IS curricula? The results presented here show that there is a continue need to place more importance in technical courses and systems management issues. As end-user computing becoming increasingly common in organizations, IS professionals are expected to be competent in the technical expertise and be able to support management in all areas of business operations in the organizations and to ensure that competitive advantage can be achieved for their organizations.

Acknowledgments

This study was completed with support from the Research Group: Information Systems in Organizations at the University of Wollongong, Australia. Financial support was provided through a research grant from the Australian Research Council.

References

[1]. Ang, A. Information Systems Curricula: A South-east Asian Perspective. *Journal of Computer Information Systems*, 1992, 32(4), 10-19.

[2]. Ang, A.Y., Winley, G.K., Fan, J.P.O. Information Systems Education: A Malaysian Industrial Perspective. *International Business Schools Computing Quarterly*, 1993, 5(2), 10-21.

[3]. Ang, A.Y. & Winley, G.K. Information Systems Education: Comparison of Views from Australian and Southeast Asian Academics. *Journal of Computer Information Systems*, 1993, 34(1), 30-36.

[4]. Ang, A.Y., Winley, G.K. & Lau, S.K. Information Systems Curriculum: A Comparison of the Views of Professionals in Australia and Malaysia. *International Business Schools Computing Quarterly, 1994*, 6(2), 14-19.

[5]. Ang, A.Y., Winley, G.K. & Lau, S.K. Towards an Information Systems Curriculum for Developing Asian Nations. *Proceedings APITITE'94*, 1994, 507-513.

[6]. Cassidy, M.J. *Education and Training Needs of Computing Professionals and Para-Professionals in Australia Vol. One: Data and Findings.* Canberra: Department of Employment, Education and Training, 1990.

[7]. Fabri, T. & Mann, R. A Critical Analysis of the ACM and DPMA Curriculum Models. *Journal of Computer Information Systems*, 1993, 34(1), 77-80.

[8]. Goslar, M.D. & Deans, P.C. A Pilot Study of Information Systems Curriculum in Foreign Educational Institutions. *Journal of Computer Information Systems*, 1993, 34(1), 8-17.

[9]. Jiang, J.J. Requisite Skills for New Business Graduates; Recruiters' Views. *Journal of Computer Information Systems*, 1994, 35(1), 28-30.

[10]. Jones, S. & Dick, G. Business Information Technology Program at University of New South Wales: Meeting the Needs of Industry? *IFIP Transactions A: Computer Science and Technology*, 1994, A-54, 331-341.

[11]. Knapp, J.A. Information Systems: Educational Offerings Versus Industry Needs. How well do they match? *Proceedings of the 1993 ACM SIGCPR Conference*, 1993, 18-26.

[12]. Monin, J.D. & Deive, P.J. Skills in an Environment of Turbulence: A Survey of Information Systems

Professionals in New Zealand. *Proceedings of the 1994 SIGPLAN Conference*, 1994, 208-218.

[13]. Quarstein, V.A., Ramakrishna, H.V. & Vijayaraman, B.S. Meeting the IT Challenges of Business: The Changing Mandate for MBA's. *Information Systems Management*, 1994, 11(2), 62-70.

[14]. Richards, M. & Pelley, L. The Ten Most Valuable Components of an Information Systems Education. *Information and Management*, 1994, 27(1), 59-68.

[15]. Sallis, P.J. Management Perceptions of IS Research and Development Issues. *Asia-Pacific Journal of Information Management*, 1994, 2.

[16]. Trauth, E.M., Farwell, D.W. & Lee. D. The IS Expectation Gap: Industry Expectations Versus Academic Preparation. *MIS Quarterly*, 1993, 17(3), 293-307.

[17]. Winley, G., Lau, S.K. & Ang, A.Y. A Pilot Study of the Tasks of Information Systems Professionals in Indonesia. *Journal of Computer Information Systems*, 1995, 35(4), 59-65.

Intelligent Slides
A Strategy to Promote Active Learning[1]

Sandeep Purao, Assistant Professor of CIS
Georgia State University, Atlanta, GA 30302-4015
Net: spurao@gsu.edu, **Web:** http://cis.gsu.edu/~spurao, **Voice: 404 651 3859, Fax: 404 651 3842**

Abstract

Effective learning requires active student participation. It is through an active synthesis of concepts that the student arrives at a deeper understanding of materials presented in class. Current lecture and course formats do not provide any overt means to encourage this activity – synthesis of concepts – a key contributor to active learning. This paper argues for creating an information-base that can provide students ample and *explicit opportunities to explore and synthesize concepts, across lectures as well as across related external materials.* The scenario envisioned is: lecture materials that could be used by the instructor (in the classroom), as well as by students (outside the classroom) to explore the subject matter. The paper demonstrates how this can be achieved using the existing software capabilities. The approach was implemented for a graduate database class at a major university. The enhanced materials were delivered to students over the world-wide-web. Use by students required a Reader, which is available free-of-cost. Final results indicate that the approach was successful in promoting active learning while protecting existing intellectual investments in slides.

Motivation

Learning is not a spectator sport. Effective learning requires active student participation [Gamson 1991]. It is through an active synthesis of concepts that the student arrives at a deeper understanding of materials presented in class. Synthesis of concepts has been recognized as a key contributor to active learning [Bloom 1954, Gamson 1991].

Current lecture and course formats do not provide any overt means to encourage this activity. Occasionally, such connections are made by some industrious students, much to our surprise and pleasure. More often though, it is the instructor who implicitly assumes the responsibility for making these connections - either through class discussions, or by way of clever exercises or assignments that require students to (i) synthesize concepts from different lectures, or (ii) explore related external materials. To ensure student participation, we believe that an information-base is needed which can promote active learning by providing students ample and *explicit opportunities to explore and synthesize concepts, across lectures as well as across related external materials.* A possible response to this problem involves organizing the lecture materials with links across related concepts that can be activated by students.

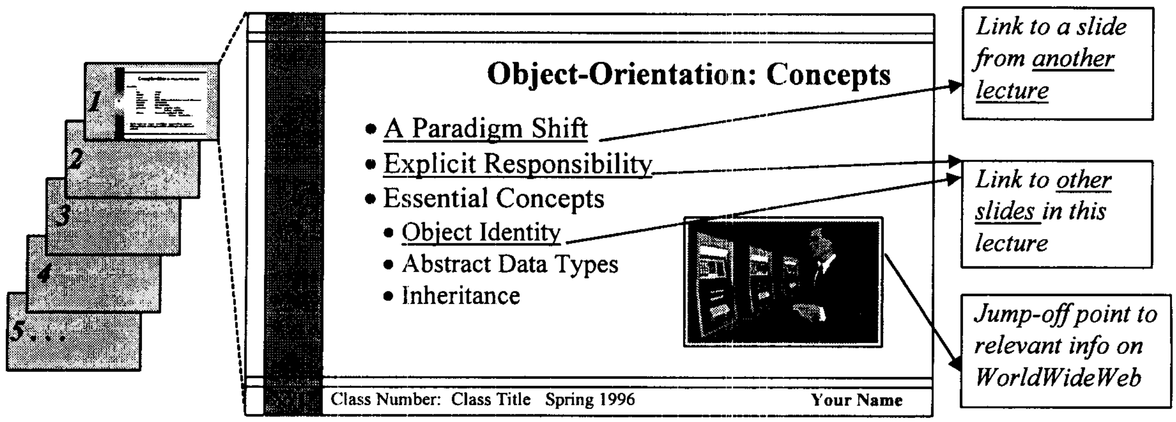
The recent practice of using the World Wide Web for delivery of course materials (such as lecture slides) facilitates *physical links*, say, between the course schedule and lecture slides. However, these links remain at a coarse level of granularity, say, between files. Semantic links across specific topics are not formed. The solution calls for an approach similar to the 'Help' facility found in many of today's software packages, that is, a *hypertext* approach that explicitly provides intelligent jump-off points that students can explore on their own. This allows links, at the appropriate level of granularity, across different but related topics.

The problems with implementing a full-scale hypertext solution are, however, many. First of course, is the effort required to implement the solution, which is magnified in a fast-changing field such as Computer Information Systems. Second, and perhaps more important is the inability to use such a 'Help' solution directly for class lectures – which remain a critical component of course delivery today. Since educators rightfully expend a considerable amount of time and efforts on developing lecture presentations, a hypertext solution that is independent of lecture preparations cannot be justified. Finally, it is also necessary that the jump-off points connect not only different topics with one another, but also with externally available resources such as those on the world wide web.

[1] *This work was supported by an Instructional Innovation Grant from the College of Business, Georgia State University, Atlanta, GA.*

Intelligent Slides

The ideal scenario we envision is create lecture slides that can be used by the instructor (in the classroom) as a basis for class discussions or presentations, as well as by students (outside the classroom) to explore the subject matter (see figure 1 below).



Slides used in class for Lectures by the instructor, and out of class for exploration by the students.

Figure 1: Intelligent Slides

The information-base would be integrated with lecture preparations to prevent duplication of efforts, and would contain jump-off points to relevant external sources. At first glance, two existing hypertext technologies appear as plausible implementation candidates for this problem. One involves constructing the information-base as a 'Help' feature. The other involves development using the increasingly popular *world-wide-web.* Though they satisfy some requirements (see table 1), neither allows easy integration with lectures – which would be critical for adoption by students and instructors.

Requirements	**Help**	***Web***
Internal Links	Yes	Yes
External Jump-off Points	No	Yes
Implementation Effort	High	Low
Integration with Lectures	No	No

Table 1: Current Implementation Candidates

It is also important to recognize the following, ancillary, requirements for a workable solution:

- the lecture materials should be viewable / printable, but should not be manipulatable, *safeguarding intellectual property.*
- the solution should preserve current investments in course preparation by allowing *leveraging of existing slides.*
- the solution should be *uniformly applicable* across packages (such as PowerPoint™ or FreeLance™) for different instructors
- the lecture materials should also be *uniformly accessible* across platforms (such as PCs or Macs) to different students

Implementation

It was realized that implementing Intelligent Slides would require imaginative use of existing capabilities of software tools. Towards this end, a few tools were explored (PowerPoint™, Visual Basic™, and ToolBook™) before settling upon the eventual choice, which provides, admirably, most of the requirements outlined above: Adobe Acrobat Exchange™. Figure 2 outlines the process of creation, delivery and use of these slides, where the shaded boxes indicate additional work relative to traditional lecture preparation.

It maintains use of lecture slides *in the classroom*, and enhances and organizes them in an information-base – with hypertext links to related concepts and jump-off points to external information sources. The information-base can be browsed by students *outside the classroom* for exploration and synthesis of concepts.

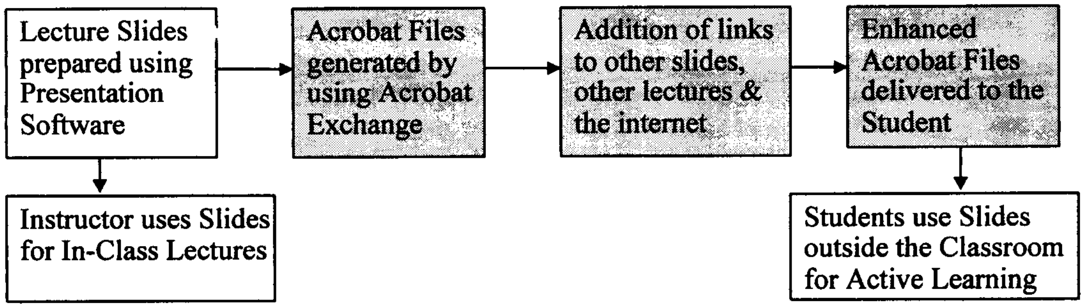


Figure 2: Creation, Delivery and Use of Intelligent Slides

The feasibility of the approach was tested for a graduate Database class at the author's home institution. A license for the software Adobe Acrobat Exchange™ is required to generate the slides. The enhanced course materials are delivered to students over the world-wide-web. Use of the slides by students requires an Adobe Acrobat Reader which can be obtained free-of-cost. Figures 3 to 8 below show snapshots of the implementation. Complete implementation is available at `http://cis.gsu.edu/cis814`.

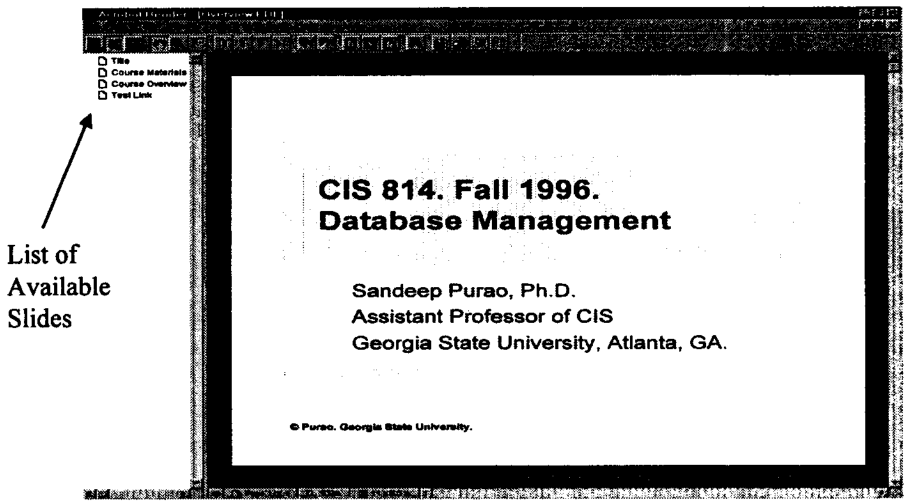


Figure 3: Initial Slide for the Course (File: Overview.PDF)

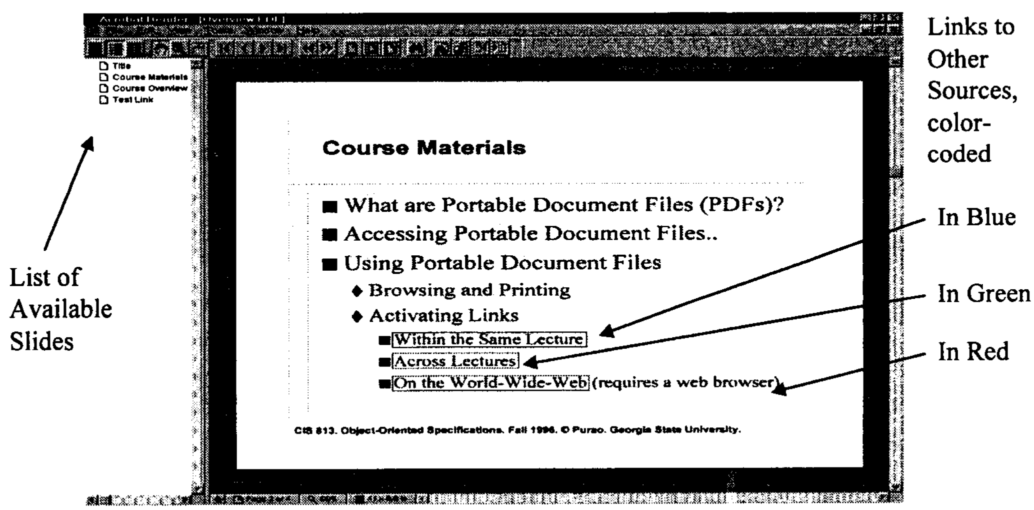


Figure 4: Explaining Course Materials Organization (File: Overview.PDF)

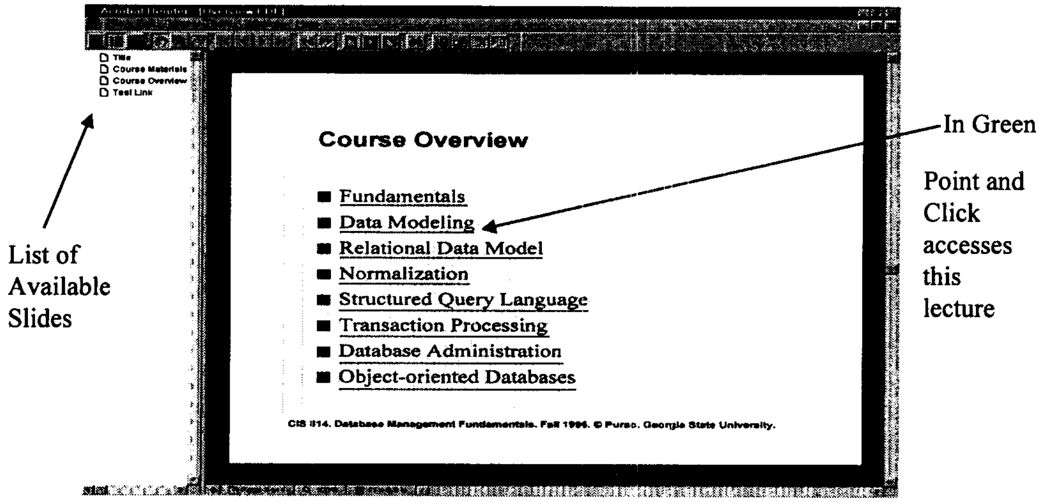


Figure 5: Course Overview (File: Overview.PDF)

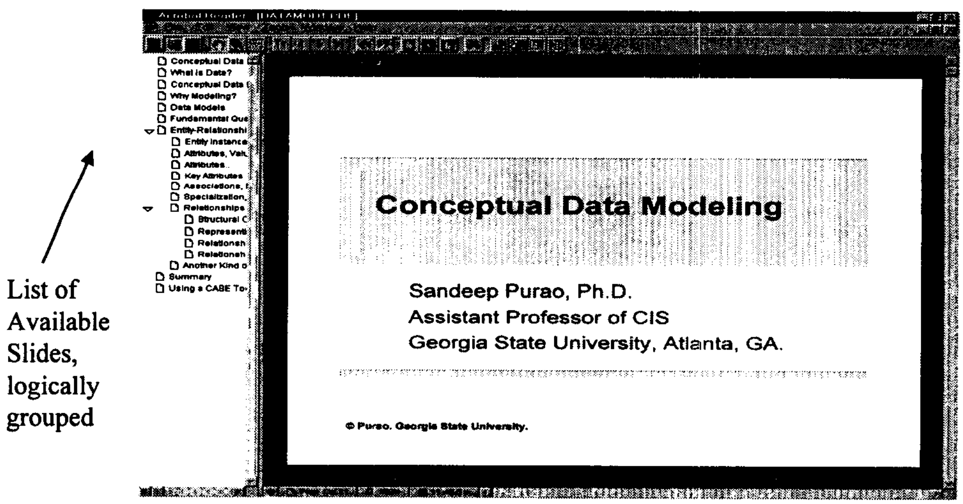


Figure 6: Lecture on Conceptual Data Modeling (File: DataMod1.PDF)

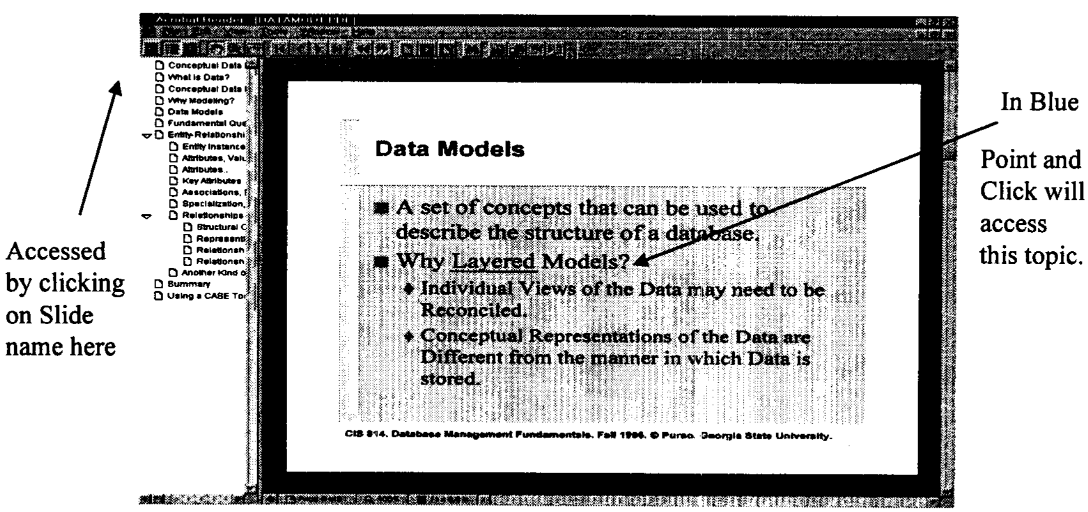
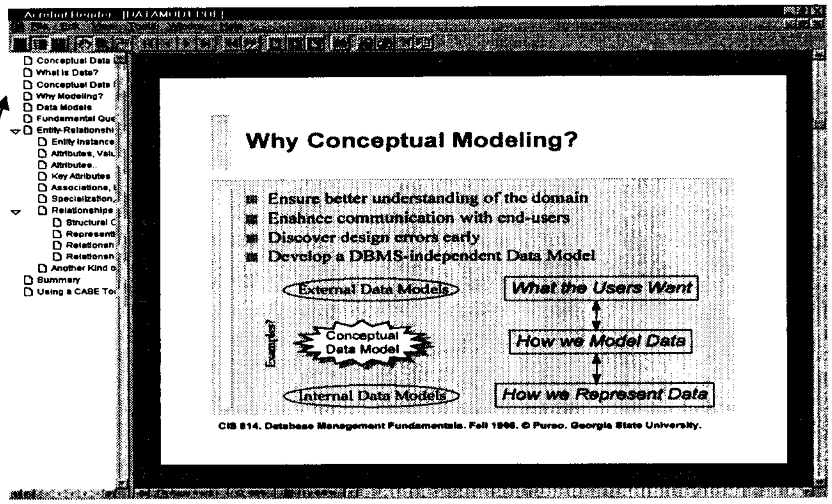


Figure 7: A Slide in the Lecture on Conceptual Data Modeling (File: DataMod1.PDF)



Slide 3 in this lecture

Accessed by clicking on Layered in the last slide.

Figure 8: Another Slide in the Same Lecture (File: DataMod1.PDF)

Results

The approach was successful in seamlessly integrating out-of-class student explorations with in-class lectures. Student response was extremely positive. Use of links was reported on various occasions: preparing for class, refreshers on earlier concepts, and seeking further information. The students easily took to using the information-base. Based on the quality and nature of class discussions, it appeared that the students did use the information-base to explore and synthesize concepts from lectures. Table 2 summarizes the benefits.

Benefits	
Internal Links	Yes
External Jump-off Points	Yes
Implementation Effort	Low
Integration with Lectures	Yes
Leveraging existing Slides	Yes
Safeguarding Intellectual Property	Yes
Effectiveness	
Adoption and Use	Easy
Exploration of Materials	Promoted
Synthesis of Concepts	Facilitated

Table 2: Benefits and Effectiveness

To recap, the pedagogical innovation demonstrated in this paper is ***Intelligent Slides***. I have described how the approach can be implemented and showed some snapshots. The approach has potential to promote active learning by providing students ample and *explicit opportunities to explore and synthesize concepts* presented within as well as beyond the lecture materials. The approach has proved useful for one graduate course. Currently, it is an integral part of lecture design by the author in multiple courses. Considering its ease of use and long list of benefits, it should be useful at both undergraduate as well as graduate levels. Educators at other schools can easily adopt the approach. It does not require the adopters to start anew. Existing lecture slides can be easily enhanced to take advantage of the approach. It is not constrained by lecture preparation software (e.g. PowerPoint™) or computing platforms (e.g. PCs, Macs). The software Adobe Acrobat Exchange™ is inexpensive – $495 for an educational workgroup (10 copies) license [Adobe 1996]. The information-base is portable (e.g. from PCs to Macs). Use by students requires two software tools – Adobe Acrobat Reader ™ [Adobe 1996] and a web browser such as Netscape Navigator™ – both can be acquired by the students *free-of-cost.*

References

1. Adobe, Inc. 1996. About the Adobe Acrobat Software Suite. On the *World Wide Web* at http://www.adobe.com/
2. Bloom, B.S. 1954. Taxonomy of Educational Objectives. Longamans, Green & Co., New York.
3. Gamson, Z. 1991. A Brief History of the Seven Principles for Good Practice in Undergraduate Education. *New Directions for Teaching and Learning.* Vol. 47, Fall. Pp. 5-12.

Laying the Foundation for a Challenging IS Curriculum

Joyce Dunston
Assistant Professor
Data Processing Unit
Hostos Community College

Julio Gallardo
Professor
Data Processing Unit
Hostos Community College

Abstract

Goal three of the 1990 National Education Goals calls for schools at all levels to create challenging curricula. At a community college in New York two information systems instructors are working to achieve this objective. In doing so, the instructors created a teaching strategy called the CAL (Concrete Active Learning) Approach. The Approach helps instructors make reading, writing, discussing and critical thinking integral parts of a course's learning experience. The Approach is designed for use at the introductory level to prepare students for the rigors of advanced studies

At a community college in the Bronx, New York there exist a movement to create a learner-centered Information System (IS) curriculum where:

- students are provided regular opportunities for in-depth investigations of key topics and problems,
- active learning, thinking and doing are emphasized;
- the development of cognitive skills and processes, such as writing, are re-enforced; and
- multiple resources and media are used for teaching and learning (1, p.4).

The curriculum revision began with the exit course, Internship, where the first step taken was the establishment of course goals. These are:

1. to expose students to the IS work environment and
2. to provide a means by which instructors can enhance and evaluate students' overall abilities as an IS professional.

With regard to the former, students are placed at various organizations where they perform the duties of an entry level IS employee. In terms of the latter, students are assigned to work in teams to solve real or simulated IS problems. The instructor overseeing the courses behaves as a supervisor guiding students as they identify the problem, design a plan of action, search for possible solution, evaluate solutions, and apply the solution(s) which is (are) most appropriate. During the experience, the instructor observes the students' abilities to demonstrate the characteristic principles of an IS graduate as defined by Longenecker, et al.(2) These include the ability to communicate in writing and orally; to think critically and solve problems; and to work as a member of a team.

Having determined the goals and learning experiences of the exit course, the next step was to revise the foundation courses. Given the students' abilities at entry into the course of study and the outcomes to be achieved, it was found that students needed a learning experience which would give them the opportunity to practice and improve basic and thinking skills while at the same time acquire knowledge of the discipline. A teaching strategy called the CAL (Concrete Active Learning) Approach was created for this purpose. The CAL Approach uses a set of well-structured activities that systematically combine three teaching methodologies: the SQ3R (Survey, Question, Read, Review, Recite) reading method, co-operative learning, and the writing process. The Approach is used in the introductory courses to prepare students for the high-order thinking assignments they will encounter in the upper-level courses. (2)

Each of the three instructional strategies, SQ3R, co-operative learning, and the writing process, brings a special benefit to the CAL Approach.

By using the SQ3R reading method students are guided in surveying and questioning reading assignments. This helps them identify the purpose for which they are reading and sets the stage for critical reading and thinking.

Co-operative learning, which forms the basis of all CAL Approach activities, requires students to perform various writing tasks in small groups. In this way "...students work together to maximize their own and each others learning." (3, p.174) Studies show that the face to face discussion of course concepts helps students gain a deeper understanding of the material. (4) According to Robert Stahl, the effective application of co-operative learning calls for the use of fourteen elements. In the CAL Approach eight of the fourteen elements are re-enforced. These are:

- establish a clear set of specific learning outcome objectives
- establish a clear and complete set of task-completing directions or instructions
- provide opportunities to complete required information-processing tasks
- organize students into heterogeneous groups
- establish positive interdependence
- arrange students so that hey have face to face interaction
- allow sufficient amount of time for students to use the activities
- ensure that each student is held responsible and accountable for doing his or her own share of the work (5)

The first three elements are built into all CAL Approach activities, as each activity segment contains:

1. a set of specific learning outcomes (including those relating to cognitive skill development);
2. a complete set of directions; and
3. reading and writing tasks that give students the opportunity to process the information

It is the instructor's role to ensure that the remaining five elements are present.

In terms of the third strategy, the CAL Approach accepts the view that writing is a process which entails three phases: planning, translating and revising. Through CAL Approach activities students complete writing tasks that are associated with the planning and translating phases of the writing process. These tasks include sorting, classifying, translating, information, reproducing, and listing. By completing these tasks students become actively engaged in the learning process and re-enforce low-order thinking skills as defined in Bloom's Taxonomy. (6)

CAL Approach activities consist of two parts: The S-Q (Survey-Question) Module and the Writing Process Module. The S-Q Module leads students in working collaboratively to execute tasks that help them Survey and Question course reading materials. Out of class, students Read the assigned materials and complete the homework assignment specified in the S-Q Module. In class (and once again working collaboratively) students use the instructions in the Writing Process Module to perform writing tasks. These tasks help students Review what has been read, then Recite (actually restate) what has been reviewed. In this way SQ3R, the writing process and collaborative learning are combined to actively engage students in the learning process, while at the same time re-enforcing basic and thinking skill development.

Below are examples of an S-Q Module and Writing Process Module.

S-Q (SURVEY-QUESTION) MODULE

The CAL Approach is a teaching methodology that combines reading, writing and collaborative learning. A complete CAL Approach Activity consists of two parts: an S-Q Module and a Writing Process Module. While it is suggested that an S-Q Module be used in concert with each Writing Process Module, it is not required. Use of an S-Q Module ensures that the SQ3R (Survey, Question, Read, Review, Recite) reading method is fully utilized. S-Q Modules are the part of the CAL Approach that helps students to Survey and Question the material to be read. Each S-Q Module consists of two parts. These are labeled In-Class and Homework.

Objective: Students examine a set of learning expectations or important topics.

INSTRUCTIONS FOR STUDENTS

In-class

You have been provided with several learning expectations and topics. Under each expectation or topic there is one question.

1. Working individually.

   a) locate in the textbook (or other reading material) the answer to the first two questions that appear on the sheet.

   b) write the answer directly as it appears in the textbook.

2. Now working with a partner compare your results.

3. If you answers differ, decide on one answer to present to the instructor.

4. Submit your answers to the two questions to the instructor.

Homework

1. Read the chapter

2. As you read, locate the answers for the rest of the questions on the sheet that was distributed in class.

3. Write your answers directly from the textbook (or other reading material).

Note to Instructor

This activity segment requires that you develop a set of learning expectations and a set of questions about the chapter. In this way, you can help focus students' attention to the areas you deem most important.

Be sure to review the answers the students submit.

At an appropriate point during the coverage of the material, have the students answer the questions again. Note the change in their answers.

WRITING PROCESS MODULE: ANALYZING LECTURE NOTES

A complete CAL Approach activity consist of two parts: an S-Q Module and a Writing Process Module. It is suggested that an appropriate S-Q Module be used prior to the use of any Writing Process Modules.

> Critical!!!
>
> Before applying the Writing Process Module, it is important that the instructor complete the two items listed within this box. Completing these items will help ensure that the activity focuses on essential content areas.
>
> Major Syllabus Topic:
>
> Specific Content to Be Learned Or Learning Expectations (include here specific learning outcomes related to the major topic above)

Instructional Function

1. Motivational hook before lecture.

2. Expand coverage beyond the information provided in the textbook.

Cognitive (Basic and Thinking) Skill Development Objectives.

Basic Skill: All Writing Process Modules of the CAL Approach focus on the development of the writing skills. This activity use the planning and translating phases of the writing process. Students reproduce definitions which they locate and use the controlled writing technique to re-write selected passages of the reading material.

*Thinking Skills: As students perform the writing tasks they

- Define new terms using a dictionary or the textbook
- Distinguish between facts that are already known from those that are not known
- Express what is not known
- Explain known items

*Adapted from Bloom, et al., Taxonomy of Educational Objectives: Cognitive Domain as per article Higher-order Thinking by Charles Miller

Directions to Students

Part I

You have X minutes to read the notes distributed to you. These notes relate to the topics on pages to of your textbook.

1. When reading the notes, do the following:

   a Read the notes to get an overview of their contents. Use a dictionary, the textbook glossary or the textbook to define any words or terms you don't understand. Maintain a list of these words or terms you don't understand. Maintain a list of these words or terms and their definitions.

   b Read the notes again. This time mark (underline, circle or highlight) the items you don't understand or about which you are confused.

   c After you have identified all the items you don't understand or about which you are confused, use the controlled writing technique to alter the passages marked. Your instructor will have explained to you the controlled writing technique.

Part II

You have X minutes to complete Pat II. Please do the following:

1. Sit with your group members

2. Each member must assume one of the roles below:

   a Manager: Keep the group on track, encourage participation and organize the work

   b Writer/Recorder: Record discussion and conclusions

   c Checker: Check material to ensure that the facts and statements are correct

   d Timekeeper: Keep the group abreast of the time used and time remaining

3. At this point, members are to take turns sharing with the group the items that were not understood in the reading.

4. When a member shares his/her item of confusion, the group should determine whether or not anyone in the group can explain the item. If yes, then he/she (or they) should do so. If no, the group should construct a question that when answered clears up the confusion or write a statement that explains what is confusing about the item.

   Remember:

   a you should consult your textbook (as well as any other materials, i.e. homework, you deem necessary) to help clarify items of confusion

   b to take turns sharing items of confusion.

5. Submit questions and explanations prepared in step 4 to the instructor.

Directions to Instructor

1. Create a set of lecture notes on a particular topic covered in the textbook. This may be the perfect time to expand on a topic for which the textbook, for your purposes, does not provide sufficient coverage.

2. Be sure to bring extra dictionaries to class for students to consult. Both a technical dictionary (subject specific) and an everyday dictionary would be useful.

3. Demonstrate the controlled writing technique to students. According to Anita Colleen controlled writing "Students copy well-written passages altering them in some significant way, such as from present to past tense, from singular to plural, or from (technical to commonly used English terms) to help them use spontaneously some of the more basic conversions of the language. (See Writing Instruction in the Two-year College. ED 272258)

4. Circulate among the groups to hear their discussions, troubleshoot problems and to address some questions. As you circulate, you might like to call the attention of the entire class to discuss salient points brought up by one of the groups.

5. At the end of the session or at a subsequent session, address some of the questions or explanations the students will submit.

6. When organizing students into groups be sure that the five elements of co-operative learning listed below are addressed:

   a organize students into heterogeneous groups

   b establish positive interdependence within the group

   c arrange students so that they have face to face interaction

   d allow sufficient amount of time for students to use the activities

   e ensure that each student is held responsible and accountable for his or her own share of the work

   f Once groups have been organized, allow students to work in the same group for periods of 4 to 6 weeks.

The format of each module, in particular the Writing Process Module, follows a modified version of the Susan Prescott Activity Chart (7). The format is designed to help instructors effectively plan their lessors. For example, the Writing Process Module contains 6 sections. By completing the first two sections, Major Syllabus Topic and Specific Content to be learned, the instructor is forced to identify those course concepts he/she feels are most salient. The third section, Instruction Function, aids the instructor in determining at what point in the lesson the activity will be used. That is, at the beginning as a motivational book; after a reading assignment to check for understanding; or in the middle as a means of providing concrete engagement prior to the presentation of abstract ideas. The next section (Cognitive Skill Development Objectives) states, in measurable terms, the writing and thinking skill objectives to be achieved. The section Directions to Students defines the work that students will do in order to meet the writing and thinking skill objectives. the final section notes to the instructor which provide helpful tips when applying CAL Approach activity.

The next step in the curriculum revision process is to create a full set of CAL Approach activities for the two introductory courses that comprise the College's IS curriculum. The Approach can then be tested for its effectiveness in re-enforcing writing and reading skills and its contribution to content acquisition. Finally, should the test prove successful, instructional materials need to be developed to aid faculty in using the Approach in the courses they are instructing.

With a solid foundation in basic and low-order thinking skills students will be better prepared to meet the ever increasing challenges they will face as they move through each level of the curriculum.

Works Cited:

1. Patrick, John. "Achievement of Goal Three of the Six National Education Goals." ERIC. May 1993. ED360221.

2. Longenecker, H. and Feinstein, D., ed. Information Systems The DPMA Two Year Model Curriculum For IS Professionals, IS 94. Park Ridge: DPMA, 1994.

3. Johnson, David and Johnson, Roger. "Implementing Cooperative Learning." Contemporary Education, Vol. 63, No. 3, Spring 1992 p. 173-180.

4. Balkcom, Stephen. "Cooperative Learning." ERIC, June 1992. ED 346999.

5. Stahl, Robert. "The Essential elements of Cooperative Learning in the Classroom." ERIC, March 1994. ED370881.

6. Miller, Charles. "Higher-Order Thinking An Integrated Approach for Your Classroom." Vocational Education Journal, October 1990, p. 27.

7. Prescott, Susan. "Trouble Shooting Cooperative Learning in the Classroom." What works: Building Effective Collaborative Learning Experiences (Conference Proceedings), National Center for Teaching and Learning

The Evolution and Re-Invention of Information Systems Curricula in Australia

Arthur Tatnall

Department of Information Systems
Victoria University of Technology
PO Box 14428 MCMC, Melbourne, Australia 8001

email: ArthurTatnall@vut.edu.au

Abstract

Courses in Computer Science were first offered in Australia's universities in the 1950s, but Business Computing (Information Systems) courses did not emerge until the mid 1960s as a response to the requirement of the Australian Commonwealth Government for computing professionals to fulfil its growing administrative needs. This paper traces the beginnings and the evolution of Information Systems curricula in Australian tertiary institutions and proposes a model of how these courses have continued to develop from the 1960s to the late 1980s. It concludes with some thoughts on how this model might project into the future.

Introduction

Courses in business computing are unusual in that they involve, in essence, the study of a particular machine and how humans use this machine to assist in operating and managing their organisations. The rapidity of change in computer technology provided the major impetus for evolution of courses in business computing, and also had an important effect on what was taught.

While it could be argued that knowledge of how courses have developed is irrelevant, and that we should be concerned only with the future, such an argument fails to recognise the importance of an understanding of historical developments in the shaping of present and future events.

Early Computing in Australia's Universities

In the 1950s and early 60s when digital computers first became generally available, Australia had only a small number of universities. Probably the first tertiary studies in computing began in 1947 in the University of Sydney's Department of Mathematics when Trevor Pearcey introduced a course in *The Theory of Computation, Computing Practices, and the Theory of Programming.* This course was offered before CSIRAC[1] (Australia's first computer) became operational in 1948 (Pearcey 1988). Prior to this time, some courses had been offered in Victorian Technical Colleges in the use of punch-card operated *accounting* machines. Whilst not *computing* courses, these did lay some of the ground work for courses in business computing, and continued well into the 1960s.

Australia's early computers were based in the universities, with CSIRAC operating in the University of Sydney from 1948 - 1956, and Melbourne University of from 1956 –1964. From the mid-1950s a number of university computer systems were opened to general use, and practical training in programming and the application of computers was introduced at the Universities of Melbourne, Sydney, and NSW. Early training courses, of a few weeks duration, were offered in the techniques of programming appropriate to each machine. (At that time, of course, to use a computer *at all* really required some knowledge of programming.) It was, however, some time before education in computing was seen anywhere other than university departments of statistics and mathematics.

Programming courses were given regularly in the University of Melbourne from 1956, and in 1959 a formal subject in *Numerical Methods and Computing* was developed by Pearcey as part of an undergraduate course in pure mathematics. Undergraduate courses in the *Theory of Computation* commenced in 1964 with the establishment of the Department of Information Science (Pearcey 1988).

Intervention of the Commonwealth: the Defence and PMG Computing Projects

In 1957, at the time of the second Australian conference on *Automatic Computing and Data Processing,* Pearcey (1988) reports that there were no computers in commercial use in this country, and the only machines in existence were in the universities and at the Weapons Research Establishment. The conference was divided into sections on Programming and Mathematics, Engineering, and Business Applications. Along with the large number of papers on topics such as programming and systems

[1] CSIRAC (or CSIR Mk 1 as it was first known) was built for the Australian Commonwealth Scientific and Industrial Research Organisation (CSIRO) by Trevor Pearcey and Maston Beard. CSIRAC is one of the earliest first-generation computers, and was probably the fifth operational stored-program computer in the world. It is also interesting to note that it was not until 1948 that Pearcey and Beard became aware of the parallel work on EDSAC (Pearcey, 1988).

design, were several papers by J.A. Ovenstone discussing commercial and administrative applications of computers.

In the late 1950s and early 1960s, partially due to the efforts of several individuals such as Ovenstone, and partially as a response to worldwide enthusiasm with systems analysis and computers, the Commonwealth Public Service, and in particular the Department of Defence and the Postmaster General's Department (PMG), began to investigate the advantages of using computers in administration. It can thus be argued that the Commonwealth Government was the pacesetter in computing in Australia during the 1960s, and that in many ways the Commonwealth Public Service, rather than commerce in general, paved the way for business computing in Australia.

In 1958, Ovenstone was appointed Controller of Automatic Data Processing (ADP) in the Department of Defence in Canberra, and set out to apply 'automatic computing methods' to the administration of the defence sector. Although implementation of the Defence Project proceeded well, a severe lack of personnel sufficiently knowledgable in computing, along with difficulties with the acquisition and installation of equipment, caused some delays. Perhaps an even bigger problem was the fact that these projects used all available qualified computing personnel, leaving few available to undertake other tasks (Maynard 1990). The need for trained staff, created by these projects, very soon had considerable repercussions on the tertiary education system.

In 1959 the Commonwealth Public Service Board (CPSB) began to investigate how best to introduce computer technology. Regarding training, its investigation found inadequacies, particularly in the training of systems analysts, and reported that many organisations considered that a four-week course by a manufacturer, coupled with on-the-job knowledge, was sufficient qualification. On the other hand, it noted that the United States Government realised that the ability to think 'creatively, imaginatively and analytically' was very important, but that knowledge of higher statistics, often taught in university courses in computing, was not seen as being particularly useful. The US Government required candidates for computing positions to have had two years experience in Organisation and Methods (O&M) and one as a systems analyst, or to undertake a six-months systems analysis course full-time. The report stressed that the Board should provide leadership in course provision and that control of training should be centralised (Philcox 1978).

Wasting little time after acceptance of the recommendations, in 1960 the Board ran its first course in *Analysis and Design of Mechanised Systems* with Ovenstone as one of the lecturers (Philcox 1978). Also in 1960, Ovenstone organised a training scheme for the Department of Defence, and the Board ran over twenty *Systems Analysis and Design* courses each of twelve weeks duration for its middle management officers. These courses involved twelve consecutive weeks of full-time training, both theoretical and practical, and were conducted by the Board at its ADP Training Centre in Melbourne. Course content included: introduction to systems analysis and design (1 week), equipment characteristics (1 week), basic programming (5 weeks) and A.D.P. systems analysis and design (5 weeks). The three-month courses continued for some years, and Maynard (1990) describes the courses he ran before 1965, whilst working for the PMG's Department, as about half systems analysis and design, and half programming.

'Unsuitability' of the University Courses

During the 1950s the universities had only been gradually coming to grips with the issue of whether computing was a part of mathematics or should be considered as a new discipline. With courses which were quite theoretical in nature, relatively few staff and sparse facilities, they were largely unprepared for the demands of the 1960s when the Commonwealth Government's Defence and PMG Computing Projects, as well as the needs of industry, produced a requirement for trained computing personnel that was massive in comparison to previous requirements.

What was needed were courses with a substantial component that was vocational in nature and the universities were then not interested in providing such courses. By 1964, the bulk of ADP training was being done outside the universities, much of it by the Commonwealth itself which was somewhat critical of the universities in this regard (Commonwealth of Australia, 1968). Computing courses at the universities, whose focus was internal, rather than on industry needs, were not providing a direct return, leaving the way open for the soon to be created Colleges of Advanced Education (CAEs) to do so.

The Programmers-in-Training Scheme

Although regarding its 12-week training as highly successful in providing 'crash courses' in computing, the Public Service Board recognised the need to set up longer courses and began the design of a twelve-months long *Programmer-in-Training* (PIT) course. The first PIT course ran in 1965 and initially drew upon the Defence staff's experiences with both computerised, and existing non-computerised, administrative systems in the Commonwealth Public Service. Maynard was involved in running PIT courses for the Board from 1965 until he joined the Department of Civil Aviation in 1969 as a senior programmer in charge of application programming.

He remained in this position for about a year before taking up a position at Caulfield Institute of Technology, in Melbourne, in 1970 where his involvement with PIT courses continued (Blackmore 1992). Maynard describes the PIT course as a 'double-decker sandwich course of one year duration combining periods of formal classroom education with on-the-job training' (Maynard 1990).

The PIT courses took over 20 hours/week of formal class time for one year and operated initially in Canberra and Melbourne, but later in other capital cities (Maynard 1990; Pearcey 1988). The forty-six week course covered the following topics: introduction to the course and the service; computer equipment and techniques 1, 2; computer mathematics 1, 2 (statistics); programming 1, 2; systems analysis and design 1, 2. PIT courses continued, under the Board until late in the 1960s when responsibility transferred completely to the tertiary education sector. These courses set the style for many of the courses later offered in the Colleges of Advanced Education and Institutes of Technology (Maynard 1990).

Courses in Colleges of Advanced Education

In 1961 a report by the Australian Universities Commission (the Martin Report) recommended the establishment of a binary system for tertiary education where the established universities of the time would remain as highly academic, research-based institutions, while Colleges of Advanced Education (CAEs) would be created from the existing Technical Colleges and Teachers Colleges. The newly established CAEs would initially be restricted to a teaching role. Rasmussen (1989) notes that prior to this report, the Victorian Technical Colleges were the most highly developed and regarded in Australia and greatly influenced the Commission's conclusions.

One of the first educational institutions in Australia to adopt computing as a priority teaching area was Caulfield Technical College, later to become Caulfield Institute of Technology. Maynard (1990) claims these courses to be the first electronic data processing courses in Australia. As early as 1961 Caulfield had offered a *Certificate of Accounting (DP)* course, and Maynard suggests that things evolved from there. Prior to 1964 the College had offered a number of short, evening, post-diploma courses in various aspects of computing such as: punched-card systems, accounting machine applications, commercial electronic data processing, and principles of analogue computing, which were offered for a small fee, and delivered (in the main) by lecturers brought in from industry (Greig and Levin 1989). Maynard reflects that in both Australian and world-wide terms, these courses were very early: "people were still teaching accounting machines - IBM punch-card machines. They were trying to move from punch-cards into the computing arena, but IBM was very slow in really getting computers into business." (Maynard 1990).

In 1970 the Commonwealth Public Service Board decided to hand over the running of the Programmer-In-Training course to four selected tertiary institutions: Caulfield Institute of Technology, Bendigo Institute of Technology, the newly established Canberra College of Advanced Education, and New South Wales Institute of Technology (Maynard 1990).

Peter Juliff took most of the programming classes in the early PIT courses run by Caulfield and describes their format as 'long short-courses'. They did not involve lectures in the normal university mould, but a much more intense six hours of class time per day, five days per week, during the three months periods the students were on campus - more like the sort of offering in the two or three day short courses offered by industry training houses today (Juliff 1992).

Courses in business computing in the CAEs thus did not evolve by diverging from university computer science courses, but developed quite independently. According to Juliff (1990) it is useful to consider the relationship between three separate computing course entities during the 1970s:

- 'Academic' Computer Science courses offered in institutions like Melbourne University,
- An Applied Sciences type of business computing course offered by Caulfield, Swinburne, Footscray, Preston and the smaller Institutes and CAEs and,
- Courses in Business degrees like those offered at Prahran, Swinburne and Preston CAEs.

Juliff maintains that all three course types have grown up together to produce the spectrum we have now. The aim of the courses was typically to turn out programmers, and the career path then was not a lot different from today: students went off course essentially as programmers, then went on to work in systems analysis and system design.

A Model for the Evolution of Courses in Business Computing

A study of the curriculum history of business computing reveals a complex interrelationship between computer technology, educational needs, institutional facilities, and the ongoing development of curricula. These elements can be represented as an overlapping cycle as shown, but one that repeats again and again. Each new cycle is initiated by a *trigger* which can either represent a significant change in technology, a new educational need, or a change in the way society views the technology (Tatnall, 1993).

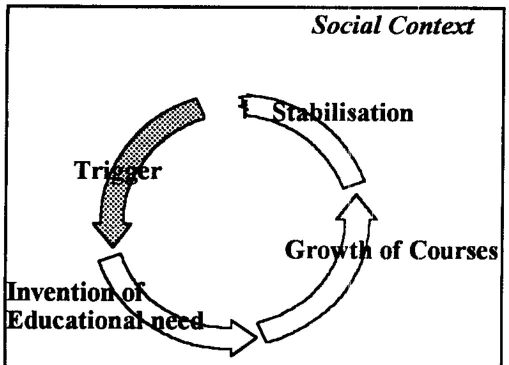


Figure 1

Part of the reason for this complexity is the rapidity of the changes in available computer technology that have occurred during this period. These rapid changes, and the relative slowness of the educational response, caused tension in tertiary institutions offering courses in business computing as they geared to move from mainframes to minis to microcomputers, from punch-cards to terminals, from the use of magnetic tape to magnetic disk, or from O&M to systems analysis. The frantic speed of development led to the culture associated with computing being given scant attention with the result that there was often a need to *re-invent* concepts in each new cycle, sometimes accompanied by a repetition of earlier mistakes.

The technology

The most obvious aspect of computer technology is the hardware itself: the mainframe, mini or microcomputer. Software is also of great importance and today it is the software package, rather than the computer it runs on, that is the critical ingredient. But application packages like word processing are a recent phenomenon. It is hard to imagine many businesses being seriously interested in generating client letters from punched-card input, and although text editors were available on minicomputers in the 1970s, few people *seriously* thought of using them for writing. It required the improved user interface and relatively low cost of the microcomputer to make word processing generally accessible. Other important changes in computer technology include the development of systems analysis and design and the structured programming methodologies in the 1960s and 70s.

Invention of a use and an educational need

Once the new computers existed in a useful form their potential for use in government and commercial organisations had to be justified to potential users, but as Franklin (1990) suggests, in the excitement accompanying the creation of a new technology, justifying a need to use it is not difficult. In the 1960s many claims that today seem far fetched, were made in an attempt to sell the need for business to use computers: they would reduce cost, improve efficiency, led to paperless offices, give everyone more spare time, and generally improve everybody's quality of life. Often the claims were couched in terms that implied that businesses not using computers would soon loose their competitive edge and cease to exist. When the microcomputer first appeared, a similar attempt had to be made to sell the need for such devices in the home, office and school. Magazines of the early 1980s are full of suggestions for writing BASIC programs to balance your cheque book, track your physical fitness, read your biorhythms, or store your recipes. Many of these ideas seem silly today, but were seriously raised at the time in an attempt to justify a need. When it was clear that use of the machines could be justified, a need to train computing professionals in their use was quickly invented as there was little point in having computers if no one could use them. In the excitement of the time, a need for education in the use of computers would have seemed self-evident.

Growth of courses and infrastructure

For growth of courses to occur there had to be an educational infrastructure - physical resources, central support facilities, technicians and teaching laboratories. This infrastructure was important as it affected the way in which subsequent developments could occur, and once set up was difficult to change or dismantle. When Caulfield Institute of Technology had developed a significant infrastructure to support the use of punched-card input devices to its computers in the early 1970s, it was most reluctant to consider an early move to the use of terminals requiring a completely different infrastructure (Juliff, 1990). The move from using terminals to micros required a similar change in educational infrastructure.

Stabilisation of courses

During a growth period courses undergo almost continuous development but, in time, a stage is reached where stabilisation begins to occur. One factor pushing courses towards a stable form is the influence of external constituencies, particularly professional organisations like the ACS[2], ACM and IEEE-CS. Although not possible in the early stages, when things settle down model curricula are developed and become readily available. Another factor which tends to move courses towards stabilisation is the need felt amongst academics and students for the improved status for their subject. The battle for disciplinary status that all forms of computing have faced is well known to academics in this field.

Repeating cycles

After a few years, a new generation of computer hardware or software emerges, a new methodology is developed, new demands are made on educational institutions, and the cycle recommences: course developers

---

2 Australian Computer Society

forget much of the course development history and start again, largely from the beginning.

There have thus been a number of education cycles in business computing - I suggest there have been five up to the mid-1980s (Tatnall, 1993). These are similar in that they were each aimed at the education of computing professionals, but different in how they tried to do this and in the technology they employed. While the start of some cycles was due to advances in technology, this was not always the case. The two most important events in the evolution of courses in business computing in Australia: the need of the Commonwealth Government for computer professionals in the early to mid 1960s, and the creation of the Colleges of Advanced Education in the late 1960s, were not directly related to changes in technology.

The model is thus composed of five of these cycles. In each case, the cycles overlap in time, some by a considerable amount, meaning that the start of a new cycle did not necessarily mean extinction of the earlier cycle. New cycles are triggered by some significant event such as the rise of a new technology, or of a new educational need. In Franklin's (1990) terms, a trigger could be considered to have something in common with the rise of a 'competing technology'.

The comparative simplicity of the model should, however, not be taken to imply that the process was either linear, or as straightforward as may appear to be suggested, and the system was far from being one closed from societal influences. A series of overlapping, triggered cycles could now be pictured as shown below.

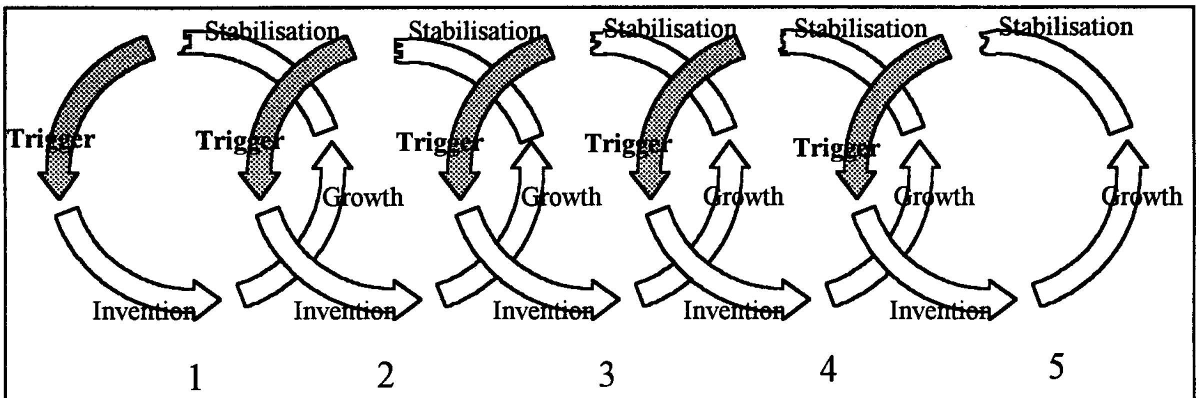


Figure 2

Trigger:	Invention of accounting machines	Invention of 1st generation computers	Needs of the Commonwealth Government	Martin Report on tertiary education	Invention of the microcomputer
Date:	1930s	1950s	1960s	1970s	1980s

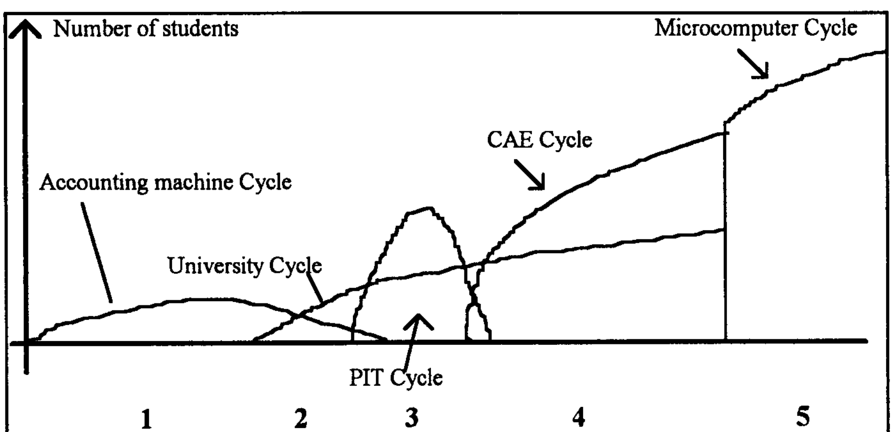


Figure 3

Other Curriculum Cycles

One major computer-related curriculum development occurring at present is the de-emphasis of technology in favour of its uses. With early computers, users had no choice but to be concerned with the internal workings and it was not possible to use a computer at all until the early 1960s without a knowledge of programming. Advances in technology have meant that such detailed knowledge is not entirely necessary. A consequence has been a huge expansion in computer use in other curriculum areas with their resulting deep integration into many of these areas.

Another question that arises is whether other aspects of curriculum development also have a similar cyclic nature. If this is the case, it is likely that the cycling would occur more slowly than was the case with business computing.

Future Information Systems Cycles?

This study of how information systems curricula evolved in Australia is based on the use of 'Curriculum History' techniques (Tatnall, 1994) as described in much of the work of Ivor Goodson (1993). These techniques are more concerned with a discussion of curricula than with detailed history.

It is always dangerous to attempt to use historical techniques for current events or to attempt to predict the future, particularly of computing. For instance, a few years ago it may have been tempting to suggest a curriculum cycle base on the use of CD-ROM technology. Some trends do, however, appear to be quite clear, and early evidence suggests the continuance of some type of cyclic process.

The World Wide Web Cycle?

Perhaps the next cycle (following on from the current Microcomputer Cycle) will relate to the interconnection of computers into networks and the consequent ready availability of large amounts of information from around the world on the World Wide Web (WWW).

The trigger for such a cycle is quite clear, and appeared a couple of years ago in the rapid connection of many university computer laboratories to the Web. Invention of a need is now also well progressed with hardly a day going by without some mention of the WWW or the Internet in the media. Growth of courses making use of this technology is currently underway, but few courses have yet reached the point of stabilisation.

There are several aspects to use of the WWW which are likely to have profound curriculum implications. Many of these may lead to a trend towards greater similarity in IS curricula around the world.

- When it is possible to easily obtain case study and reference material from the Web, this is likely to affect what is offered in IS courses. If, by modifying a course a little, better use can be made of available material, this is likely to happen.
- Of courses this adaptation of curricula to available resources happens now in relation to text books, the future of which must be regarded as uncertain.
- The availability, on the Web, of curriculum materials prepared by IS academics from other universities, possibly from the other side of the world, may lead to a reduction in local curriculum development. When someone else has already done the work, why should a hard pressed course developer not use it?

Conclusion

Almost all of the public predictions made concerning the future nature and uses of computer technology made over the last fifty years have turned out to be wrong, some spectacularly so. Nevertheless, as long as computer technology continues to develop as it has, and business, although also changing, continues to require a supply of computing professionals, it is likely that Information Systems curricula will continue to evolve.

References

Blackmore, Tony (1992); *Professional Computing*, July/August 1992, Sydney.

Commonwealth of Australia (1968); *An Introduction to the Commonwealth Public Service with Particular Emphasis on the Growth and Control of Automatic Data Processing*, Canberra.

Franklin, Ursula (1990); *The Real World of Technology*, Montreal: CBC Enterprises.

Goodson, Ivor (1983); *School Subjects and Curriculum Change*, Croom Helm, London.

Greig, Jack and Levin, Pearl (1989); *Computing at Chisholm: The First Twenty Five Years, 1965-1989*, Chisholm Institute of Technology, Melbourne.

Juliff, Peter (1990), (1992); *tape-recorded interview*.

Maynard, Gerry (1990); *tape-recorded interview*.

Pearcey, Trevor (1988); *A History of Australian Computing*, Chisholm Institute of Technology.

Philcox, Richard, (1978); *The Commonwealth Public Service Board and the Introduction of Computer Technology, Early Australian Computer Systems*, MA Thesis, University of Melbourne. Published in *Computers and Society*, Deakin University, 1989.

Rasmussen, Carol (1989); *Poor Man's University*, Footscray Institute of Technology, Melbourne.

Tatnall, Arthur (1993); *A Curriculum History of Business Computing in Victorian Tertiary Institutions from 1960 – 1985*, MA thesis, Deakin University, Geelong.

Tatnall, Arthur (1994); *The Beginnings of Business Computing*, APITITE94 conference, Brisbane, published in proceedings.

Validating a Cognitive Flexible Hypertext Learning Aid to Teach a Structural Model of Implementation.

Alejandro Ramírez
Department of Decision Sciences and MIS
Concordia University
ramirez@alcor.concordia.ca

Abstract: The notion of Cognitive Fit was used in a Pilot Study to validate a Cognitive Flexible Hypertext Learning Aid to teach a Structural Model of Implementation. An assessment regarding the "match" between task and tool is a way to validate the use of that tool -or set of tools- in that particular context. The task, in this study, was to learn an IS implementation model. The tool was the Cognitive Flexible Hypertext Learning Aid. The test contains three levels of complexity. Test and Tool were validated. Preliminary results indicate that we are in the right track.

The Cognitive Fit Paradigm

Vessey (91) developed the notion that complexity in the task environment will be effectively reduced when the problem-solving aids support the task strategies required to perform that task. She termed that notion: "Cognitive Fit." Newell and Simon (72) stated within their Information Processing Theory that human problem solvers will strive to reduce their efforts while solving a problem, since they are limited information processors. In order to facilitate the problem-solving process that human problem solvers use in completing a task, the processing effort must be reduced. If the tool is matched to the task, the processing efforts will be reduced. This is precisely the Cognitive Fit paradigm.

In the discussion of her findings, Vessey (91) claims that matching the problem representation to the type of task to be solved results in improved decision-making performance. Using the cognitive fit paradigm, she proposed that "*problem-solving with cognitive fit results in increased speed and accuracy of performance*." Accordingly, the possibility arises to use such a construct to assess the "match" (Venkatraman, 89) between any given task and the tool — or set of tools — used to perform that particular task. An assessment regarding the "match" between task and tool is a way to validate the use of that tool — or set of tools — in that particular context.

Cognitive Fit, as a cost-benefit characteristic, suggests that if the problem representation and any tools or aids employed support the strategies (methods or processes) required to perform the task, then a more effective and efficient problem solving occurs. (Vessey, 91). This means that the problem representation a problem solver uses must be considered in the context of the task to be solved. Also, designers should concentrate on determining the characteristics of the task that the problem solvers must address. Once these characteristics have been determined, they should be supported with the appropriate tools. For example, Vessey and Galletta (91) found that graphs are more appropriate tools when the task involves spatial problem-solving elements while tables are better suited if the task involves symbolic problem-solving elements. In other words, despite individual preferences, some tasks demand specific tools; If those tools are available the task's goals will certainly be attained. This does not mean that there is only one way to obtain the task's goals. What it implies is that when cognitive fit exists, it has performance advantages.

Cognitive fit, then, can be used as a yardstick to measure the match of several task-tools pairs in different domains, and in different contexts. It will provide enough confidence to researchers, designers, or practitioners to assign a particular tool in order to perform a given task.

The Task: Learning an IS Implementation Model.

Lucas et al. (90) developed a structural model of implementation that incorporates many of the results of past implementation research studies. It is based on the research traditions of causal modeling, attitude modeling, and innovation process modeling. They use this approach to explain the phenomena of systems implementation with a conceptual model rich in theoretical implications but complex. Its complexity has been a deterrent to its inclusion in the Information Systems (IS) Curriculum.

The variables included in the model are based on the research findings of previous implementation studies. The model consists of two separable sub-models, the *user model* and the *manager model*. The logic of the model is that implementation begins with management initiation and acceptance of a given IS and ends with user satisfaction with the system. Factors leading to manager acceptance are personal (*Manager Decision Style, Manager Demographics*), task related (*Manager Job Characteristics*), and system-specific (*Manager Knowledge of System, Manager Assessment of System and Support*). *Top Management Support* is seen to influence both *Manager's Belief in the Systems Concept* and *Manager's Involvement* with systems development. Beliefs and involvement do not directly lead to *Acceptance*; rather stronger belief leads to more involvement. Also, both stronger belief and more involvement lead to more knowledge of the system. All this to say, that some factors depend on other factors which increases the complexity of the model.

Implementation Models, a Complex and Ill-structured Knowledge Domain.

The ultimate objective of implementation models is to provide guidelines for the management of implementation. Research on implementation proceeds from an exploratory stage that sets a conceptual foundation, through the definition of variables and relationships —sometimes called dimensions and factors— to the integration of these dimensions, factors and their relationships into a testable model.

Most of these models are known mainly by researchers and practitioners. Seldom, if ever, are they taught at undergraduate or master level. Exclusively, scholars pursuing a doctoral degree have the opportunity to learn them. Their complexity and ill-structuredness are a deterrent. For the future generations of IS professionals, these models are totally unknown, although these future professionals most likely will analyze, design and implement information systems. They will learn about implementation when they realize that, after the analysis and design, the system has to be implemented.

Traditional learning typically treats these categories or constructs as regular and well structured. These models are depicted as taxonomically well-defined (in terms of types, variables and structures.) However, the occurrence of these constructs in the real world tends to be more ill-defined, since they may exhibit a variety of characteristics in different situations or in different systems. Definitions often convey well-structuredness, but in real world practice, events and objects are more ill-structured. Implementation is an ill-structured domain mostly because no general rules or principles exist that can describe or predict most of the cases, there are no defining characteristics to determine appropriate action, and some dimensions are inconsistent between cases, certain aspects of cases are differentially important in different contexts, and each case appears novel because of the interaction of the different factors (Spiro et al., 87, 88.)

Instructors, concerned about serious misunderstandings of these complex models, decide not to include them within the learning material in advanced IS courses. As a justification, they relate their experiences about students' failure to appropriately transfer previously acquired knowledge into more complex issues. Among others, there are two ways to overcome the complexity issue in teaching these models. One is to simplify them by making generalizations or by reducing the number of factors, dimensions or their relationships. Another is to use a new approach in teaching them by using innovative teaching methods. The present pilot study deals with the second option: the use of Hypertext/Hypermedia as a learning aid in teaching an implementation model.

The Tool: A Cognitive Flexible Hypertext Learning Aid.

Based on Lucas et al. (90)'s Model, the *manager model* was implemented in Hypertext considering Spiro et al. (88)'s Cognitive Flexibility Theory (Ramirez 97.) The Hypertext module consists of nine screens and ten windows. Each dimension (Manager Belief in System Concept, Manager Knowledge of the System, Manager Assessment of System and Support, Manager Involvement, and Manager Acceptance) is presented on a screen. These screens contain their definition and links to the factors or dimensions that have a direct influence on them. Through the links, the windows containing the information of the factors and influencing dimensions are presented when a link is selected. Links are filtered through programs that highlight the relevant information concerning the selected factor or influencing dimension; this is one of the powerful characteristics of hypertext that provides reinforcement in the learning process. The screens also contain linear links through labelled arrows that allow the user to traverse the information sequentially.

This design, besides providing reinforcement in the learning process, directly uses hypertext's cognitive power. As mentioned previously, by showing available links and nodes on each screen, learners visually construct the hyper web which is a representation of the expert's mental model. Also, by revisiting a given factor or dimension, at different times, from different variables with different highlighted information, associations will become apparent. This is some kind of a cognitive "click" which allows learners to "know" that they have learned something new.

The module also includes five linked icons: *Dimensions, Close All Windows, Help, Notes* and *Exit Door*. With the *Dimensions* icon, users can revisit the dimensions' screen containing a brief explanation of the model and links to each dimension. The *Close All Windows* icon allows users to close all open windows and keep on working in the active screen. *Help* opens a window containing a list of the possible actions to take given a particular situation. The *Notes* icon opens an online file where users can type some notes. Once the online file is open, a sixth icon is presented: the *Printer* icon allows users to print the contents of the online file, if there is a printer available. Students may quit the application using the *Exit Door* icon.

All these elements are put together to allow learners to experience the model actively instead of passively. Learners can move back and forth within the model to "discover" the relations between dimensions and factors. These relations also help the understanding of the dimensions and factors themselves.

The tool was designed within the philosophy of constructivism. One of the assumptions that works well in this model is that meaning varies with how the individual creates meaning from his or her experiences. Unless learners choose to traverse the information sequentially, no two learners will have the same experience. Each of them will create her or his own construction of the same implementation model. That does not mean one will be right and the others wrong; the model is the same, just the understanding of it will be different.

An important feature of the learning module is that it allows learners to selectively visit each dimension and the factors/dimensions that have a direct/indirect influence on them. This option provides a way to show the pattern of association (by means of the highlighted information) which reveals the conceptual information of the model. Then, by exploring, in a constructivist manner, students are actually learning.

The main limitation of using hypertext in learning has been determined to be disorientation (*lost in hyper space.*) To prevent this problem, the module contains the *Dimensions* icon, which takes students to the first screen of the module. Therefore, every time a student needs to revisit the list of dimensions, wants to visit a specific screen, or does not know what to do, by clicking on the *Dimensions* icon, he/she will be able to get back to a "safe" location. The title of each screen refers to the dimension under discussion. The *Help* icon provides information on the options available any time it is requested. With all these aids, disorientation is unlikely.

Validating the Tool: The Pilot Study

Within the notion of Vessey's (91) Cognitive Fit, where complexity in the task environment (acquisition of advanced knowledge in a complex and ill-structured knowledge domain) will be effectively reduced when the problem-solving aids (hypertext-based learning aids) support the task strategies required to perform that task. This pilot study is part of an empirical research addressing the following questions as an extension of the concept of cognitive fit:

- *Do hypertext-based learning aids enhance the acquisition of advanced knowledge?*

or to put it in other words,

- *Will learning in a hypertext-based learning aid result in increased speed and accuracy of performance?*

and

- *Are there gender differences with respect to student performance when using hypertext and linear text materials?*

In order to answer the previous questions, a controlled experiment was designed. Its goal is to measure differences between the tools, and if so, which one of them is better suited for that particular task. As part of this experiment, a pilot study was conducted to validate the tool. In this study the task was: Advanced knowledge acquisition in a complex and ill-structured domain. There were two different tools: A computer-based linear version and a computer-based Hypertext version of the Lucas et al. (90) Implementation Model.

Methodology

Subjects were thirty-six (58.33% males, 41.67% females) MIS students enrolled in an *Information Systems Design* course. Participation was evaluated as a Lab Project. It represented 5% of the course grade. Subjects were randomly assigned to one of two experimental treatment groups: *Group L* (Linear) worked in a computer-based linear version of the learning material; and *Group H* (Hypertext) worked in a computer-based hypertext version of the same material.

The instructional domain is one with which university-level students majoring in MIS are assumed to have some familiarity because of previous exposure to formal course work in the field. The topic is of interest, we hope, to these students and thus motivates them to participate enthusiastically in this study.

Procedure

There was an introduction session at the beginning of the pilot study for all of the participants. In this session subjects were informed about the experiment, of the two versions of the learning material, also that they had been randomly assigned to one of those groups. They were informed that they could spend as much time as they like working with the material.

Subjects had a hands-on training session. The training material basically shows subjects how to move around the system. Only when they were familiar with the system were they exposed to the learning material. Students could spend as much time as needed to learn the content.

A computer-based test consisting of three levels of complexity was used to evaluate performance. Complexity 1 consisted in six multiple choice questions regarding the comprehension of the dimensions of the model. These questions deal with basic understanding of the material. Complexity 2 consisted in a short case and two multiple choice questions regarding the synthesis of the model and the case information. In this level, subjects are asked to make inferences regarding the model. Finally, Complexity 3 consisted in a longer case and five multiple choice questions regarding the evaluation of the information on the case using the model. The case contains information on an information system and the people in the organization that have an influential role in the implementation of it. The players are introduced with a list of facts regarding the variables in the Lucas et al. (90) model. In addition to the test, students completed an evaluation form at the end of the session.

Data that was obtained from this pilot study includes the time spent in the learning module, the time to complete the test, the option selected for each question, the scores for each level of complexity, and a total score for the test. These data were kept in a record including information of the task type (hypertext/linear) and gender of the subject (male/female.)

The evaluation form contains twelve Likert scale keys associated in six factors: Participation, content, task, test, hardware, global satisfaction. Three factors had sub-factors. Content included motivation, usefulness, and experience. Task included motivation, satisfaction, and user friendliness. Test included grade of difficulty, challenge, and motivation.

The term validity, when applied to educational material, refers to the precision with which this material measures some cognitive ability. There are thus two aspects of validity: what is measured and how precisely it is measured. The cognitive abilities referred to are abilities to perform observable tasks. How precisely a test measures an ability is indicated by the reliability of the scores. Reliability is a necessary condition for validity. Though it is not a sufficient condition (Ebel & Frisbie, 86.)

Reliability uses the correlation coefficient as a measurement. One of the properties of the correlation coefficient is that it provides a relative, rather than an absolute, measure of agreement between pairs of scores for the same persons. As a measurement of reliability the correlation coefficients were calculated for each test question, level of complexity, time on module, time on test, and each key of the evaluation form.

Preliminary Results

The first level of analysis contrasted the association of each level of complexity with total score to see whether they are equally associated. Their correlation was significant at the 0.05 level, but Complexity 2 was not significant at the 0.01 level. The next level of analysis contrasted each level of complexity with its questions. For Complexity 1 all the correlation of its questions are significant at the 0.05 level; only the correlation for question 1 was not significant at the 0.01 level. For Complexity 2 the correlations of its two questions were significant at the 0.01 level. For Complexity 3 all correlations but the one for question 12 were significant at the 0.05 level. Only the correlations for questions 9, 11 and 13 were significant at the 0.01 level.

From these results we can conclude that the reliability of the test can be improved by reviewing mainly Complexity 2, in other words, case 1. Also, the reliability of Complexity 3 may be threatened by question 12, that particular question can be eliminated.

Analyzing the time spent in the module and in the test, only their correlation with Complexity 1 were significant at the 0.01 level. Which indicates that the cognitive skills needed to answer correctly those questions profit by the time spent both on the module and on the test. Which makes sense since, at this level, memory plays a major role. On the other hand, time on module does not help the cognitive skills needed to synthesize or evaluate.

Regarding the evaluation form, significant correlations at the 0.01 level with global satisfaction include participation, content-motivation, content-usefulness, task (all three sub-factors: motivation, satisfaction, user friendliness), and test-challenge. It is important to notice that for these subjects, hardware didn't have an impact on global satisfaction. Also, from the lack of significant correlation with test-difficulty and test-motivation, we can assume that these subjects didn't find the test difficult or the cases particularly motivating. By rewording parts of the cases, the latter can be addressed. Finally, a lack of significant correlation between global satisfaction and content-experience confirms that the learning material was new to these students.

Even though the goal of the pilot study was to validate the tool and the test, we used the test results to explore whether the data shows any indication about the questions that are guiding this research. We ran T-tests of degree of complexity by task (hypertext/ linear.) These results are not conclusive but the significance (2-tailed) increases when complexity does. **(Complex 1 p=0.712, Complex 2 p=0.165, Complex 3 p=0.072)** Results of T-tests of time-in-module and time-in-test by task (H/L) both indicate that there is no difference. In the pilot study time-in-test didn't specify time in each one of the three complexity levels of the test, we will modify the tracing program to indicate them. The only significant difference on T-test regarding the keys of the evaluation form, was task-user friendliness, *Group H* found the module more user-friendly than *Group L.*

When groups were broken down by Task-Gender (HM, HF, LM, LF,) there were significant differences on score in the sub-groups LM and LF **(t_{16}= -4.142, p=0.001)** and in the sub-groups HM and LM **(t_{19}= 2.710, p=0.014)**.

Implications for Future Research

Hypertext is not the solution for every problem in learning. There are many aspects of learning where hypertext does not have a meaningful contribution. What is needed then, is some criterion to determine which aspects of the learning process are better served with hypertext. Those areas where information is not linear, or where by imposing linearity the information is not oversimplified, are excellent choices.

The new hypertext learning aids should foster learning by discovery in a constructivist way, bring some kind of structure to ill-structured material, and allow material to be revisited at different times, in different contexts. If the amount of information is overwhelming, it should be broken into modules and meaningful links should allow the traversal of each of them from different parts of the system. If necessary, the system should provide "guided tours" to the information. One aspect to investigate on these guided tours, is whether they promote motivation to learn.

The next step in our research will be to test the benefits of learning using the hypertext module. We will conduct a controlled experiment to measure differences, if any, in performance in a problem-solving situation. We expect to find evidence to support the claim that performance will increase proportionally to the complexity of the problems to be solved by those subjects using the hypertext learning aid.

One aspect that should not be ignored in future research, is the fact that motivation to learn something plays a major role in learning and training. Once the desire to learn something arises, the *how*, *when*, and *where* are almost irrelevant. Until that desire becomes second nature to students and trainees, one way to overcome some of their problems in learning is to keep looking for new methods and innovative techniques that help them. To test them effectively, one aspect that should be stressed is to motivate them to use these new methods and techniques in an almost *natural way.*

References

Ebel, R. L. and Frisbie, D. A. "Essentials of Educational Measurement." 4th. Edition. Prentice-Hall, Inc.: Englewood Cliffs, NJ.

Lucas, Jr., H.C., Ginzberg, M.J., and Schultz, R.L. "Information Systems Implementation: Testing a structural model." Ablex Publishing Co.: Norwood, NJ. (1990)

Newell, A. and Simon, H. "Human Problem Solving" Prentice Hall, Englewood Cliffs, NJ. (1972)

Ramirez, A. "The Design of a Cognitive Flexible Hypertext Learning Aid to Teach a Structural Model of Implementation." In *Proceedings of ED-MEDIA 97, the World Conference on Educational Multimedia and Hypermedia.* AACE Press: Charlottesville, VA (1997.)

Spiro, R.J., Vispoel, W.P., Schmitz, J.G., Samarapungavan, A., and Boerger, A.E. **Knowledge Acquisition for Application: Cognitive Flexibility and Transfer in Complex Content Domains.** *Technical Report No. 409*, University of Illinois at Urbana-Champaign. (1987)

Spiro, R.J., Coulson, R.L., Feltovich, P.J., and Anderson, D.K. "Cognitive Flexibility Theory: Advanced Knowledge Acquisition in Ill-structured Domains." In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society.* Lawrence Erlbaum Associates: Hillsdale, NJ. (1988,) 375-383.

Venkatraman, N. **The concept of fit in strategy research: Toward verbal and statistical correspondence.** *Academy of Management Review*, 14, 3, 423-444 (1989).

Vessey, I. "Cognitive Fit: A Theory-based Analysis of the Graphs versus Tables Literature." *Decision Sciences*, 22, (1991), 219-240.

Vessey, I. and Galletta, D. **Cognitive Fit: An empirical Study of Information Acquisition.** *Information Systems Research*, 2, 1, (1991), 63-84.

Cobol Curriculum 2000

Ronald Kizior, Loyola University
Wilson Price, Object-Z Publishing

For the past several years, those of us teaching Cobol have witnessed an alarming migration away from Cobol. What's happened? The hot job markets of the 1980s cooled off. With that word out, enrollments fell off drastically. Of course, with diminishing enrollments, institutions cut back on their Cobol offerings, in many cases, dropping Cobol altogether. Unfortunately, the loss of interest in Cobol has far exceeded the facts of life. Why? What happened? To answer that, consider the following.

- Cobol is dead. It's behind the times (the last standard was in 1985); it's verbose; its clumsy; and so on.
- C, C++, and Java are where the money is; they're the future.

There's a certain amount of fact and fiction to both of the above. (Sadly, to a large percentage of students entering the programming field, the above is perceived as 100% factual.) Cobol is dead—absurd. Behind the times—true. In this rapidly changing field, it's inconceivable that we have a span well in excess of *10 years* between standards. But at least we have a *true* standard. Verbose—true. But that's one of its pillars of strength. The money's not there with Cobol—that's history, the going rate for Cobol programmers has exploded.

As we are seeing now, the perceived extinction of Cobol has turned around! Two current factors are having a significant impact on the Cobol programmer market and the traditional Cobol-oriented CIS curriculum. First, the turn of the century with its Year 2000 problem has turned the business computing world up-side down. Combine that with the retiring Cobol workforce, and you see a significant upswing in the need for entry-level Cobol programmers. The experience of Lowell Socolofsky of Iowa-Western Community College is typical. The local demand for Cobol-trained programmers far exceeds the number graduates from his Iowa-Western program.

The second factor pertains to the technological evolution of software. The use of object technology has shown huge improvements in program reliability and programmer productivity. Unfortunately for the Cobol-oriented shop, the only option has been transitioning to other languages, a costly and time consuming choice. Fortunately, the next COBOL Standard includes full object-oriented capabilities. (Although the Standard has not yet been officially released, several sofware companies provide Cobol systems that conform reasonably well to the Standard's Working Draft.) The new Standard will allow the data center manager to invoke powerful object-oriented features while remaining with his/her reliable COBOL base. Furthermore, at least one Cobol vendor (Micro Focus) is marketing software that allows us to program Web applications using Cobol.

What are the implications to those of us with CIS programs centered around Cobol.? If one listens to academic circles, one will hear a variety of opinions including the following.

- Drop Cobol as a requirement and replace it with Visual Basic, Smalltalk, Java, and/or C++.
- Maintain the focus on structured methodology; object technology re: business programming, is another fad. (Remember when CASE tools were going to be the "magic bullet?) We should continually upgrade our structured methodology offerings and incorporate features of the next Standard when the Standard is finally published.
- Maintain Cobol as the language of choice but begin a transition to object orientation now by integrating it into the present course structure while maintaining an emphasis on structured methodology.
- Rebuild the Cobol sequence immediately with object orientation *the* focus, dropping the structured approach in the process.

It is the intent of this panel to explore our variety of alternatives. The session will be chaired by Wilson Price, an established Cobol teacher and author who has recently published a textbook on Object-Oriented Cobol. The remainder of the panel is comprised of experienced teachers of Cobol, each of whom has introduced object orientation into his/her classroom at some level. Each panel member will describe his/her experience and offer suggestions on the road we should take. Equally important will be input from the audience. To that end, ample time will be allocated for general discussion and the airing of any individual's specific views on the topic. It is the objective of the panel to begin the process of establishing guidelines for the future of Cobol in the CIS curriculum—that is, Cobol Curriculum 2000.

http://gise.org: A Website to Promote IS Education

Eli Cohen and Elizabeth Boyd
Grand Valley State University
eli_cohen@acm.org
betty_boyd@acm.org

Abstract

The website, http://gise.org, (Global IS Education) is devoted to improving the quality of IS Education around the world and to improving the acceptance of the study of IS education as a legitimate area of inquiry within academia around the world. This paper describes the origins and evolution of the site, its current state of affairs in terms of form and content, and the direction its organizers will take with it in the future. Currently the site provides educators with easy access to syllabi, papers, journals, and other on-line resources. This paper serves to solicit input and participation from educators in the development of the site.

Origins and Evolution

Why would any professor in his or her right mind try to improve IS education, not just on the home campus, but around the world? There is no academic reward for doing so on most campuses. It is costly, in terms of both time and money. And it detracts from those items that college administration most rewards, the publication of academic research based on the collection and analysis of data, no matter how arcane or insignificant.

It is precisely because of this challenge that the authors set out in their quest to change the status quo. There were two precipitating factors that led to this decision.

First, it became clear to the authors that there was a need for a website by which IS professors could link into the works of their colleagues no matter where they were located geographically. There was a need particularly by the colleges with the least resources for this equalizing influence.

The authors became aware of the need while traveling around the world, examining the state of IS education. They met with colleagues in Fiji, New Zealand, Australia, Hong Kong, Singapore, Malaysia, Thailand, Cypress, Israel, Finland, and Sweden. As might be expected, the greatest need for IS education support was in the institutions that were the most geographically remote and least well funded.

One source of support for IS education are the several fine journals that provide articles that deal with IS education. However, these journals have limited readership. While the purpose of these publications is to share information, they are constrained by their reliance on paper-based publishing. The economics of paper-based publishing is a cruel Catch 22 for academic journals. Their need to charge for the printing, just to cover the costs of printing, keeps them from being read.

While our profession may continue to use paper-based journals for the foreseeable future, it is our hope that the GISE site will widen the dissemination of information on IS education.

A second influence on the decision to create a website to support IS education was the unfulfilled hope that someone else would take on this task. About this time, the web presence of ISWorld came to the IS community. Even though ISWorld set out to provide coverage of IS education as well as research, the educational coverage at its start was scanty. Even today its coverage is quite limited. (Yogesh Malhotra developed the Business Researchers' Internet site about the same time for similar reasons.)

The authors wrote to the organizer of ISWorld, asking him to consider making the site more useful to colleagues in countries that do not have fast access to the Internet. The heavy use of unnecessary image maps for navigation was unnecessarily burdensome on scarce Internet bandwidth. Also, ISWorld's design hops the reader from site to site with most of the downloaded material being a replication of the boilerplate imposed by its design. At that time, the ISWorld placed greater importance on experimenting with leading edge technology than on providing access to colleagues whom do not have leading edge Internet support. Thus, there remained a need for a web site that promoted IS education.

Design Considerations

ISWorld is designed in a rigid, hierarchical manner. All elements of design are to be the same from page to page. It makes heavy use of and reliance on large graphic files. Responsibility for content is divided and subdivided in section editors, page editors, and such.

The design considerations at this point were to develop pages that can be downloaded in full as opposed to making a reader wander through various page links in hopes of finding something useful. In some regions of the world, it make take 15 minutes or more to download a large page, but once downloaded, it could be saved to a local disk.

Secondly, graphics were to be limited and not required for access to any content. The few graphics that were used would be reused, since downloaded graphics are cached and do not need to be downloaded again. Re-use also provided for a consistent look and feel. The initial screen of the site is shown as Figure. 1.

We confess that at first we too, like all creating their first web pages, experimented with all the bells and whistles we could, including java, javascript, midi, and active-x. These features just slowed down access and bombed web browsers and so were removed.

One design feature we kept was the use of frames. Frames enable us to keep the menu items always in view, no matter what page is being displayed in the main screen.

Originally, the web server was located at URL http://www.acm.org/~eli_cohen/globalis.htm, on the ACM computer.. Our agreement with ACM provided only 5 MB of storage. This would suffice if the site were to be limited to links to other sites.

At the start of this year, we moved the site to a server and gave it its own domain name. The server has dual T3 connections with T1 backup three hops to the backbone of Internet. This move provided us with the disk space to provide actual content in addition to links.

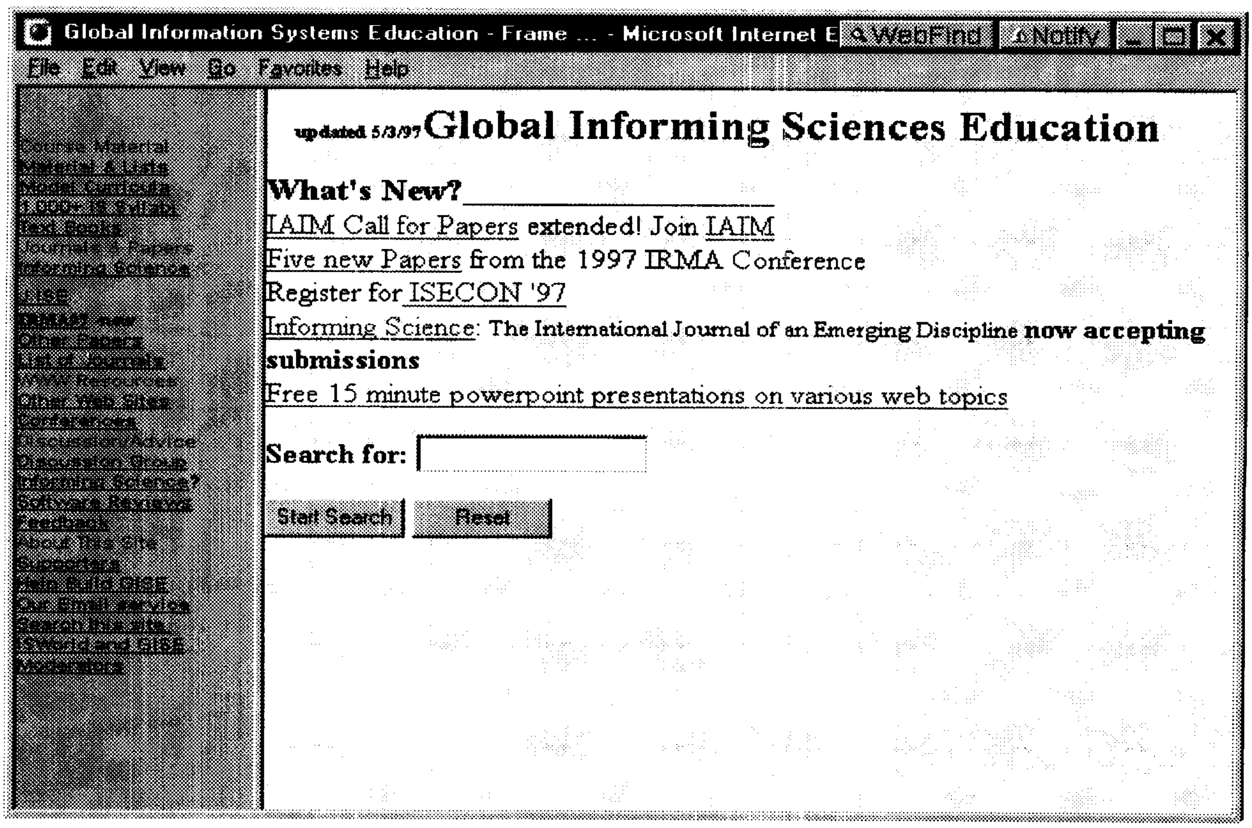


Figure 1. The Main Screen

Content Considerations

We collected links to syllabi that colleagues had placed on-line. Currently, the site links to over 1,000 syllabi.

Some suggest that we only link to the best syllabi in each of the course areas. That suggestion sounds good, but we need to ask, best for whom? A superb syllabus for a course taught at the junior level in a school that limits admission to the brightest would not be the best for an open-admissions community college. Also, we do not wish to get into the business of judging achievements that colleagues view as worthwhile.

Secondly, we sought out and established links to other material useful in improving IS education. We linked to papers offered on-line and found links to resources useful in teaching.

Work in Progress

We placed on-line a number of issues of the Journal of IS Education and, more recently, papers on IS education from the Information Resource Management Association. We intend to add more and more issues of JISE on-line, with the help of colleagues such as Sanjeev Phukan.

In addition, we have links to model curricula in our area. Also, we hope to augment our links to publishers with lists of texts for colleagues to examine. Outside of the US and in remote areas of the US, book representatives are not entirely effective in spreading the word.

Our hope is that site will house comments from those who adopt texts as to what they were looking for in a text and how well their adopted text met their needs.

What Needs Work

In addition to adding links to texts, the site needs many more improvements. For a while, the site was able to link colleagues around the world to collaborate on IS education research. Somewhere along the line, the discussion group software broke down. Our new webmaster, Geoff Mitchell of Griffith University (Australia), hopes to fix this problem. He is also working on cgi to enable colleagues to post links to their own syllabi. The proposed new look for the site, shown as Figure 2, does away with frames and uses dynamic HTML creation.

Currently we have almost 600 people who subscribe to receive an email monthly on updates to the site.

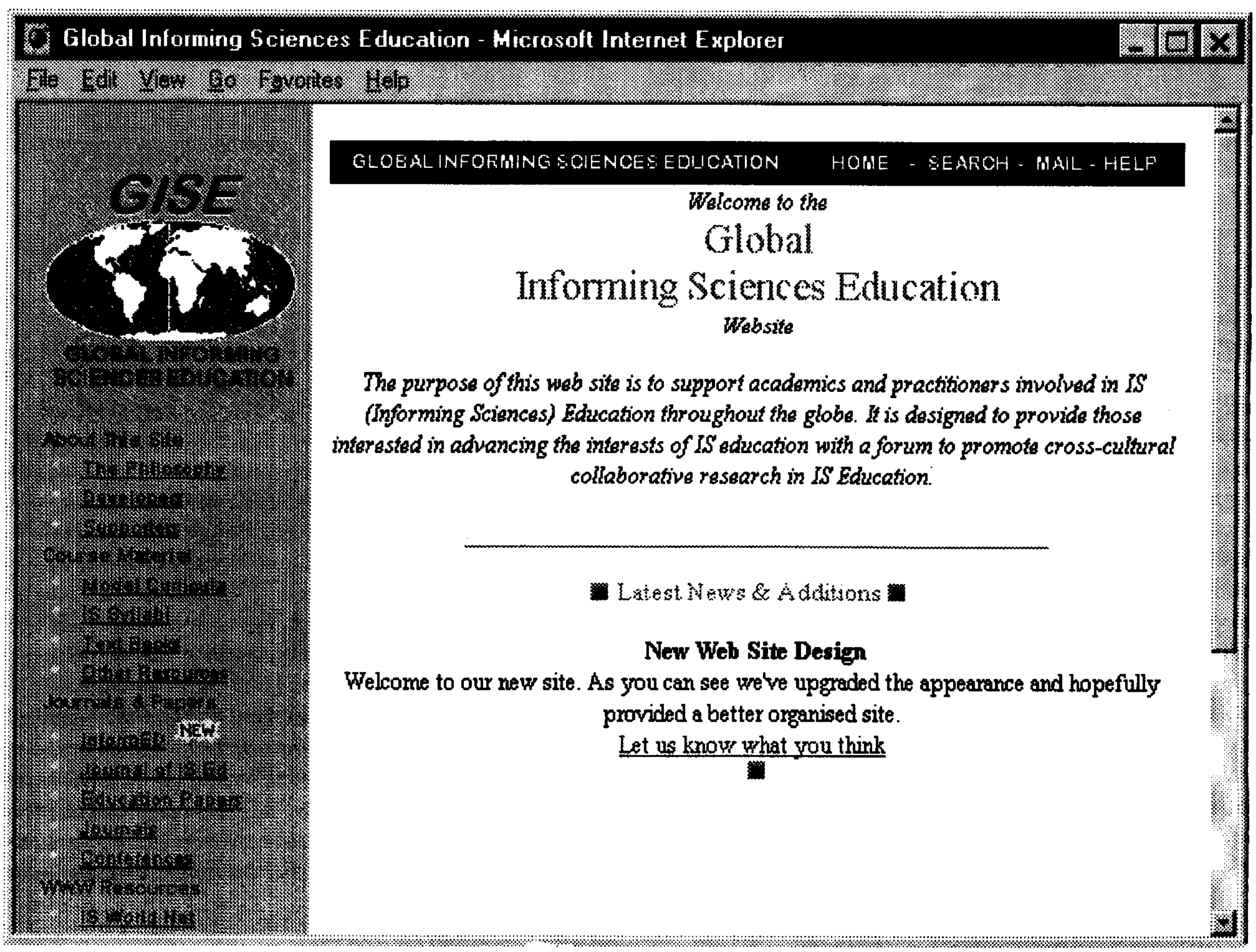


Figure 2. Proposed Design for GISE

This list processor is offered through the offices of Netiva Caftori. However, we are now looking to make the list more personal and customized to the needs of the individuals.

Also, we are actively seeking other journals and proceedings to either place their IS education papers on-line and provide us with the link or to let us post their papers.

The purpose of the site is to make resources on IS education accessible around-the-globe 24 hours a day. More than 50 educators visit the site on average each day. The actual number is likely quite a bit higher since the site is mirrored or proxied in Australia and Israel.

through corporate sponsorship. We do not currently plan to sell advertising or limit the site to those who pay.

To add value to the site, it will publish on-line the printed journal "Informing Science: The International Journal of an Emerging Discipline." We have enlisted a truly international board of editors to solicit and review articles for publication, as shown in Figure 3.

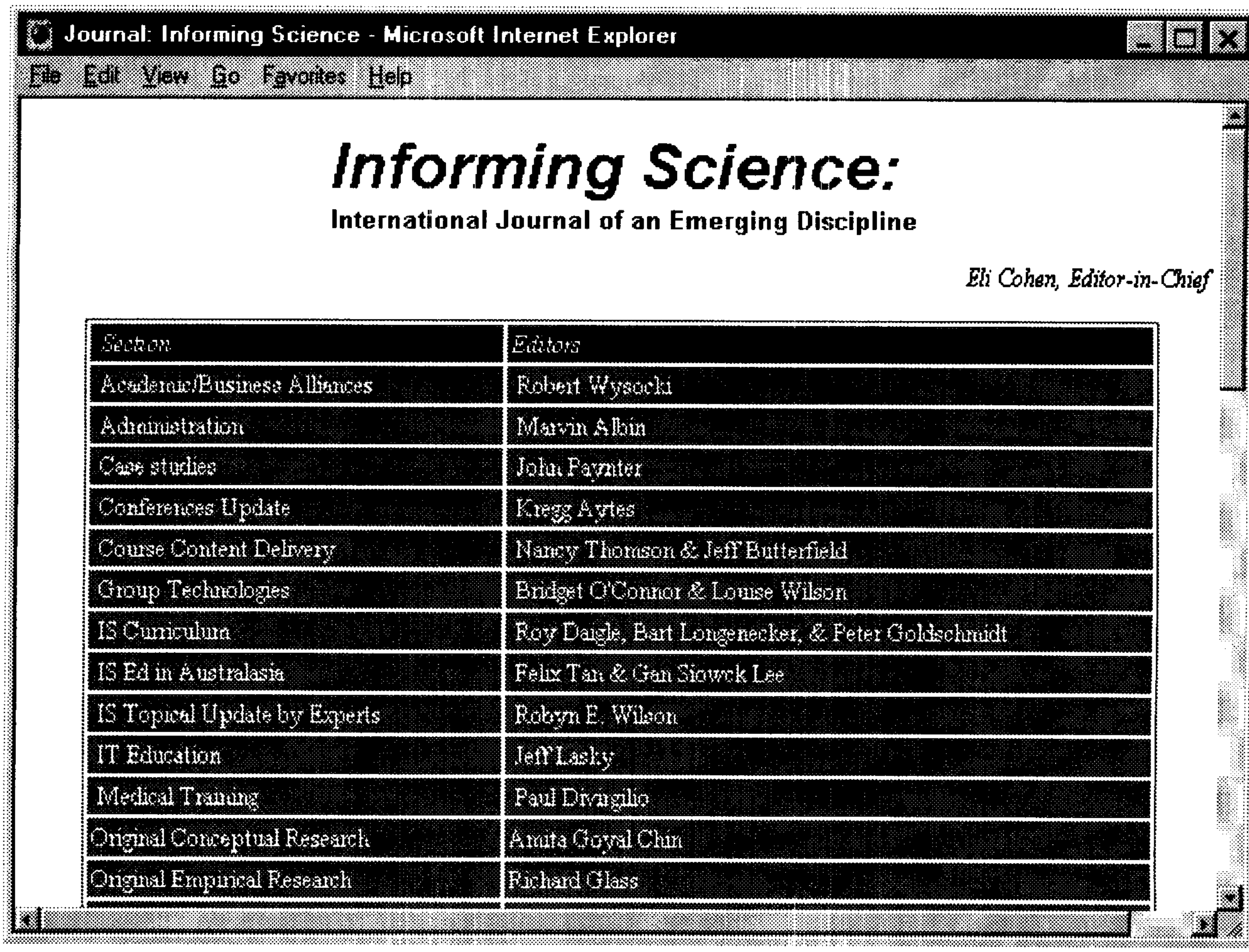


Figure 3. From <http://journal.gise.org>. Some of the Editorial Board for Informing Science

Currently it is financed through the private donations of its founders and a corporate donation from Computerworld to cover part of the first two years' operating expenses. We hope to make it self-sufficient one day

The need for Awareness of the Growing International Dimensions of Information Systems Curriculum

John D. Haney
Business Department
Walla Walla College
College Place, WA 99324
Hanejo@wwc.edu

Abstract

The emergence of a global economy and the transformation of industrial economies into information based economies has created a need for a global perspective within the information systems curriculum. The increasing development of global information systems necessitates an increasing awareness of the inseparable bond between business and information systems. The range of the curriculum response to international issues is quite expansive. This is reflected in the broad coverage of global issues in Management Information Systems textbooks. Although the curriculum is changing, the global perspective of the leadership of academic business entities has not yet seen clearly the essential integration between business and information technology. The increasing connectivity of multinational corporations, along with the trend toward electronic commerce, compels an increasing awareness of the inseparable bond between business and information systems.

Introduction

The trends of a shrinking world, in terms of diminishing barriers to global markets, have important implications for the curriculum of information systems (IS). Technology that now provides a means to transact business worldwide, or to connect a globally dispersed company, has made the internationalization of the curriculum essential. There are various approaches that will enhance a global perspective within the IS curriculum. The primary means is through the Management Information Systems course. Through this broad survey class a general perspective of how information systems relate to global business can be presented. More technical courses such as Telecommunications can also provide some international components but to a lesser degree.

The international content has expanded in the IS curriculum, but not uniformly. The range of global content from one academic institution to another can be extensive. But even given the wide expanse of international business (IB) coverage among schools, the coverage of issues with a global nature is increasing within the curriculum.

The focus of this study is the apparent lack of awareness on the part of the leadership of academic business entities concerning the growing internationalization of the information systems curriculum.

A survey of the membership of the Association of Collegiate Business Schools and Programs (ACBSP) was conducted to evaluate global business in a broad perspective. The results of part of that survey is presented as it relates to the international business content within computer related courses. Specifically, the results of the survey focus on the lack of awareness on the part of the leadership of academic business entities.

The need for the Internationalization of Information Systems Curriculum

The need for a global perspective within the curriculum is of a necessity. As (Deans & Kane, 1992) point out, there are two major forces transforming businesses today. The first is the emergence of a global economy and the other is the transformation of industrial economies into information based economies. In short, corporations are evolving into multinational enterprises.

This growth spans several levels. At the lowest level there exists exporting of unsolicited orders from abroad. Next are direct sales, where a firm's own employees market the company's products in a foreign country? Then direct production, where investment in overseas production facilities increases the level of investment and commitment in the foreign country. Full autonomy is the next stage, where a fully autonomous subsidiary in a foreign country controls the entire business in that country. And finally, a fully mature multinational enterprise would be at the global integration stage.

The issues involved are considerable: multiple languages, multiple cultures, multiple currencies, multiple time zones, multiple geographic entities, multiple regulations, and multiple governments (Deans & Kane, 1992). The common thread that runs through all of these issues is the need for information systems that provide a stable network of communication for the leadership of these multinational enterprises to maintain control.

From the management perspective there are three main concerns that surface. The first is improving customer service. The second is a closer tie between business and the information technology professionals. The third issue is understanding and using new technologies (Rayner, 1995).

The current movement toward transacting business via the World Wide Web further emphasizes the growing connection between business and technology. Tim Ouellette's (1997) article in Computer World highlights this reality. He states, "Mainframe shops are coming to grips with the fact that they soon will be invaded by probing Web browsers." And Microsoft's push for enterprise development through the Web interface with Visual Basic and C++ (PCWeek, 1997) leaves no room for doubt that a major part of the future for business transactions will be Web based. All of these factors underscore the need for the internationalization of information systems curriculum.

Review of Management Information Systems Textbooks

One of the areas of interest is whether the Management Information Systems textbooks have kept pace with the reality of the internationalization of information systems. Figure 1 lists a comparison of selected textbooks.

Comparison of MIS Textbooks		
Author	**Year**	**International Business Content**
Parker	1989	5 pages, a partial chapter
O'Brien	1996	44 pages, a full chapter
Laudon	1996	32 pages, a full chapter
Turban	1996	27 pages, within cases
Alter	1996	15 pages, spread throughout the textbook
Gordon	1996	5 pages, a partial chapter
Stair	1996	2 pages, a partial chapter
Frenzel	1996	0 pages
Licker	1997	0 pages

Figure 1

Charles Parker's (1989) textbook "Management Information Systems" is used as a base. In 1989 Parker's textbook included five pages devoted to international topics and was included in one chapter along with other issues.

As a comparison, eight textbooks that were published in 1996 or 1997 were reviewed. It was assumed that there would be a substantial increase in the international content in these textbooks. For some of the textbooks this was the case. But, as Figure 1 clarifies, there is a wide range of coverage of global issues in these textbooks. Two of the texts have entire chapters devoted to international issues, one has a section with international cases, one has international examples spread throughout the text, two have a brief component within one chapter (similar to Parker's Text in 1989), and two do not have any international content at all.

As this comparison clearly presents, the coverage of international issues within MIS textbooks spans a broad spectrum from entire chapters to no subject matter at all. Given the increasing amount of global activity by Businesses and the trends of technology, the wide expanse of this range is noteworthy.

Survey of Academic Business Entities

During the winter of 1997, a survey of the membership of the Association of Collegiate Business Schools and Programs (ACBSP) was conducted. The primary objective of the survey was to measure the Globalization of the business entities in regard to degrees, courses, course content, and the international growth of the faculty and students. A portion of that survey is utilized for this study. The part that is of interest is the component regarding the courses with international business content within the business curriculum.

Given the fact that the majority of the Management Information Systems and Computer Information Systems degrees are within business entities, the assumption was that many of the MIS classes would be identified as having an international component within them. Of the total number of questionnaires (452) sent out, there were 111 institutions that responded. These numbers are depicted in Figure 2. Of these responses, the head of the business unit filled out 85.

The question under consideration states, "List courses that include International Business topics. " Of the schools that responded there were 250 courses listed, that contained International Business topics. This is a 2.25 average per school.

Of these courses, there was only one course whose course title was computer related. This was a Management Information Systems course. It must be noted that some responses were general in nature. These responses included "all business courses," "almost all business courses," "international business in most courses," and "virtually all." So among these responses it is conceivable that the intent to include the MIS course was present.

Survey of ACBSP Membership			
	2-Year	**4-Year**	**Total**
Questionnaires			452
Respondents	45	66	111
Filled out by Administrator	34	51	85
Courses with International Business content	85	165	250
Average International Business course per school	1.89	2.50	2.25
Computer Courses with International Business	0	1	1

Figure 2

Figure 2 also provides a view of these responses broken down by 4-year and 2-year institutions. Figure 2 identifies 66, 4-year institutions with 165 courses with international business content, giving an average of 2.50 courses per school. There was only one computer related course identified. Figure 2 also identifies 45, 2-year institutions with 85 courses that have international business content, with an average of 1.89 courses per school. There were no computer-related courses identified.

There are two revelations from the analysis of this data. The first, substantiated by the international business content in the MIS textbooks as shown in Figure 1, is that the actual focus of international issues in many schools is minimal or possibly non existent. The data of this study, however, can not substantiate the actual scope of course content. Given the range of global issues covered in the MIS textbooks, it is a safe assumption that many MIS classes do provide a moderate to substantial coverage of global issues. It is conceivable, and even probable, that some schools do not teach global concepts although they are included in the textbooks.

The primary discovery of this study is the fact that the leadership of the academic business units, while filling out the questionnaire, did not consider the content of the computer classes as contributing to International Business. The most prevalent class that would provide this coverage is the MIS class. Referring to Figure 3 it is seen that of the 66 respondents that were 4-year schools, 51 of the questionnaires were filled out by an administrator. Thus, the leadership of the academic business unit filled out 77% of the surveys. Yet, based on the results of the survey, only one respondent made a connection between international business and any computer related classes.

As underscored by Rayner (1995), managers of multinational corporations see the need for increasing the ties between business and information technology professionals. They also sense a need for a greater understanding and use of technology. Historically, business and computers have been linked academically for many years. The development of business information systems has always been an integral component of the curriculum of Management Information Systems and Computer Information Systems degrees. Seemingly the administration of academic business units have not yet made the conceptual linkage between business and computer technology in their international perspective as it relates to the curriculum of information systems.

Summary and Conclusions

The emergence of a global economy and the transformation of industrial economies into information based economies has created a necessity for a global perspective within the information systems curriculum. In addition, the perspective of managers of multinational organizations envisions a need for closer ties between business and information technology professionals. They also recognize the need for a greater understanding and use of new technologies. Therefore, the Globalization of the information systems curriculum is essential.

The results of this study point out two areas of consideration. First, although the curriculum is changing to meet the growing global issues, the curriculum responses are varied and cover a broad range. This is represented by the wide spectrum of coverage of international business issues in the Management Information System textbooks.

The primary discovery, however, relates to the global perspective of the administration of academic business entities concerning the curriculum of information systems. The heads of the business entities filled out 77% of the responses to the survey. Yet the statistics indicate that these administrators have not yet made a conceptual linkage between the curriculum of information systems and the issues of international business. This is shown by the fact that out of the 165 courses that were listed with international business content, only one computer related course was listed.

The increase in the connectivity of global information systems, along with the trend toward electronic commerce must be reflected in the information systems curriculum. Awareness on the part of the leadership of academic business units of the inseparable bond between business and information systems is imperative.

References

Alter, Steven, (1996), Information Systems. Menlo Park: The Benjamin/Cummings Publishing Company, Inc., pp. 40, 86-87, 134, 235, 277, 316, 355, 409, 450, 492, 536, 583, 627, 667.

Coffee, Peter, (1997), "Visual Basic, C++ 5.0 Get the Job Done Fast," PCWeek, February 17, 1997, Volume 14, Number 7, pp. 1, 14.

Deans, P. Candace, & Kane, Michael J, (1992), Information Systems and Technology. Boston: PWS-Kent Publishing Company, pp. 1-11.

Frenzel, Carroll, (1996), Management of Information Technology. Albany: Course Technology, pp.

Gordon, Steven R. & Gordon, Judith R., (1996), Information Systems. Forth Worth: The Dryden Press, pp. 456-460.

Laudon, Kenneth C. & Laudon, Jane P., (1996), Management Information Systems. New Jersey: Prentice Hall, pp. 719-750.

Licker, Paul S., (1997), Management Information Systems. Fort Worth: The Dryden Press, pp.

O'Brien, James A., (1996), Management Information Systems. Chicago: Irwin, pp. 492-535.

Ouellette, Tim, (1997), "Web-Enabling the Enterprise," Computerworld, February 24, 1997, Volume 31, Number 8, pp. 1, 107.

Parker, Charles S., (1989), Management Information Systems. San Francisco: McGraw Hill, pp. 799-803.

Rayner, Bruce, (1995), "All Roads Lead to IT: the Global 100," Computerworld, May 1, 1995, pp. 6-8.

Stair, Ralph M., (1996), Information Systems. Albany: Boyd and Fraser, pp. 530-531.

Turban, Efraim & Mclean, Ephraim & Wetherbe, James, (1996), Information technology for Management. New York: John Wiley and Sons, Inc., pp. 775-801.

Threaded Live Case Study, Weaving Practice Through IS'97

Les Waguespack, Ph.D.

Computer Information Systems Department
Bentley College
175 Forest St., Waltham Massachusetts 02254
LWaguespack@Bentley.edu

Abstract

The case study (a.k.a. "Harvard Case") is a widely respected and utilized pedagogical instrument in management education. Although commonly used in upper level IS courses relating to strategic planning and policy, it is not so commonly used in IS courses relating to technology capability and application. This paper presents a discussion of the case study concept as it may be adapted to serve pedagogy across segments of an IS curriculum. It differs from the Harvard case in two ways: first it is live, engaging students in IS practice and second it is threaded intertwining a series of IS courses covering various curriculum aspects. The threaded live case study can help achieve greater depth of coverage and application of IS practice. It can improve the students' preparation for professional practice. The paper discusses disciplinary and resource tradeoffs that challenge the teaching of IS in a business school. The threaded live case study is described as a means of overcoming limitations resulting from these tradeoffs. The concept is illustrated in the context of IS'97 as interpreted in the BSCIS program at Bentley College and as a possible vehicle for even broader integration – across business disciplines.

Introduction

One of the characteristics that typically distinguish IS and CIS from MIS academic programs is a commitment that graduates possess high quality and mature application development skills. One of the characteristics that distinguish IS and CIS from CS academic programs is that graduates possess application domain knowledge grounded in the business disciplines (e.g. management, accountancy, marketing and finance). The business application domain enfolds the study of systems, analysis, modeling and design. The challenge for IS and CIS programs is to steward the limited program credit hour resource available in most business colleges and to optimally engage students in practical experiences. And by doing so we strive to expose students to the realities of applying best of breed IS practices in real world business situations.

The computing industry, professional organizations and institutions of higher education are committed to preparing high quality IS professionals. Their commitment is evidenced by their investment of time, people and energy in the Information Systems Curriculum Guidelines, IS '97[1] (earlier drafts of which were called IS '95). IS '97 witnesses to the importance of information system professionals to the domestic and global economies. It addresses higher education's responsibility to prepare future generations of IS professionals grounded in business domain knowledge. The sheer breadth and depth of knowledge in business and information technology described in IS '97 pose a daunting challenge to business schools in general and IS curriculum designers in particular.

This paper presents a discussion of one aspect of the overall challenge. That aspect is the task of providing IS students an effective breadth of experience in business domain based information system analysis, modeling, design and implementation. The threaded live case study is presented as a vehicle for integrating domain knowledge in the IS curriculum. Case study based pedagogy is not a new concept[2]. It is widely respected and utilized in management education. And although commonly used in upper level IS courses relating to strategic planning and policy, it is not so commonly used in IS courses relating to technology capability and application. In this paper I borrow from the case study approach and focus it on experiential learning and expand its scope to envelop a series of IS topics and / or courses. I hope that the discussion presented here will stir interest and new development of the case concept in the IS education community.

I first discuss IS education in the business school environment. I contrast the emphases of IS education with those of computer science, management information systems and general business education. How institutions deal with these competing academic interests dramatically

[1] IS'97, 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, Spring 1997.

[2] Christenson, C.R., Garvin, D.A. & Sweet, A. Eds., *Education for judgment: the artistry of discussion leadership*, Harvard Business School Press, 1991.

affects the resulting product of graduates and how they are perceived by recruiters of IS professionals. I describe a pedagogical device, I call the threaded live case study, and illustrate its proposed application in a recently implemented interpretation of IS '97, the BSCIS'96[3] curriculum at Bentley College with a short discussion of the costs and merits of the threaded case study approach. Finally, I propose a broader application of the concept that may be used to integrate content from sister business disciplines.

IS Education in a Business School

IS education continues to search for an effective and efficient means to integrate and teach knowledge from two basically orthogonal disciplines: business and information technology. Baccalaureate and graduate IS programs most often reside in a college or school of business. The breadth and depth of business education which are usually required are only found there since the business disciplines are generally ignored in colleges of liberal arts or science. Housed in a school of business there results a continuing "tug of war" between information technology and general business interests. This system of competing interests is often at the root of many institutional debates on the role of IT in business and the seat of IT curriculum in the institution.

Consider the competing interests relating to IT and business in the following learning emphasis matrix (figure 1). The rows depict the interests of information technology and business.

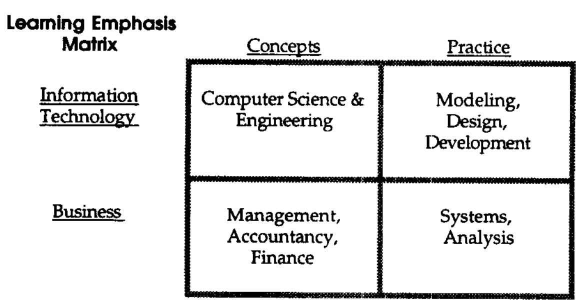


Figure 1 - Learning Emphasis Matrix

The columns represent the learning emphasis partitions of concepts (sometimes referred to as theory) and practice. Although depicted in the matrix as discrete cells, in reality the boundaries between the cells are really gradients where the interests overlap or interleave.

We introduce a graphic effect of making the border of a quadrant either thinner or thicker to depict lesser or greater emphasis on that area of influence in an academic program.

[3] Waguespack, L.J, Chand, D. R. & Robertson, J.D., "Crafting A Business Bachelor of Science Program in Information Systems Following on IS'95", Proceedings of ISECON'96, October 1996, St. Louis, MO.

Then by permuting the emphases we can consider common types of IT related programs in higher education.

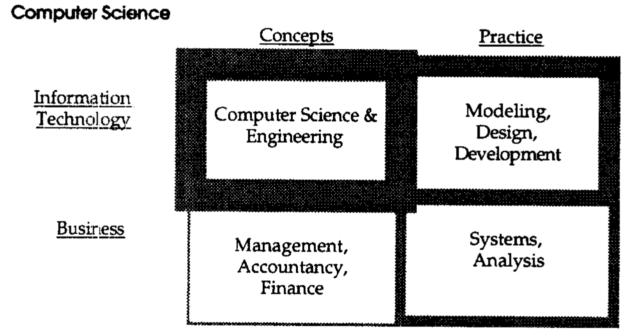


Figure 2 - Computer Science Emphasis

In figure 2 the matrix shows a heavy emphasis on computer science and engineering followed in intensity by modeling, design and development. At an even less intense level systems and analysis weigh in. Finally, business content (probably limited to management) is included. This mix of emphasis is common in computer science programs. The treatment of business domain knowledge is virtually absent. Even the theory of systems and the methodologies of analyzing systems are only minimally present. Modeling and design are usually found only in courses relating to software engineering. Development is the primary (if not the only) system cycle phase that receives any theoretical attention. Recruiters of computer science graduates often report that although these students have been equipped with great technical capabilities, they are usually ignorant of how and where to apply them in the business organization. As new hires they may be not only non-productive. but "dangerous" in business application development.

If the emphasis is shifted to the business domain quadrant the matrix shifts toward a typical business major. (See figure 3.) The three other quadrants are usually not absent, but their treatment is heavily weighted toward the information technology consumer's perspective with little if any treatment of the core IS or IT concepts.

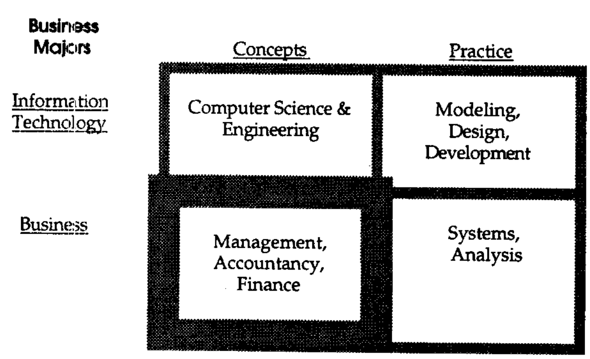


Figure 3 - General Business Emphasis

The information technology consumer perspective is a growing area of business program interest. Most business colleges recognize and have consciously encouraged an increased use of information technology in their coursework. The importance and impact of information technology can not be overlooked in coursework focusing on functional areas in business. The quality of the technology treatment, however, is primarily that of a tool in the functional area, not as an enabler of change or an opportunity for evolving the function itself. These programs ensure the graduate has awareness of IS only.

There are two hybrid models which bridge between the computer science and business matrices depicted above. The first is a variation on the business major with an increase in the emphasis on systems and systems analysis concepts. (See figure 4.) These program curricula are often ambiguously named "management information systems," MIS. The ambiguity rests in the question, "Is it systems of management information or is it management of information systems?" These programs often expand the emphasis on modeling particularly in the area of information for decision support functions in business. These programs tend to place little, if any, emphasis on system design or application development. Thus they shy away from the implementation end of the system cycle. Any computer science and engineering content that may be present is limited to issues of capacity and / or relationship rather than architecture or theory. These programs ensure graduates know what IS does, but do not prepare graduates to perform any significant IS function.

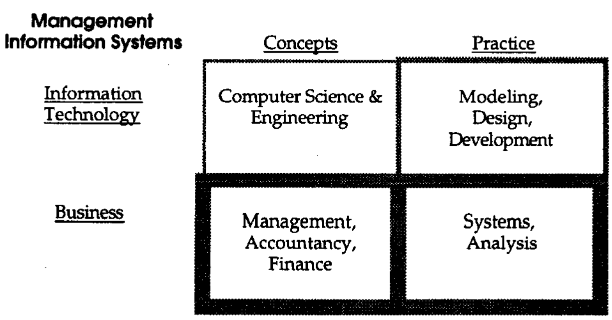


Figure 4 - Management Information Systems Emphasis

Education programs labeled "management information systems" usually offer fewer than a half dozen or so courses. MIS teaching staffs are often a half dozen or fewer faculty in a business college. College's which choose the MIS model for their technology treatment have found this to be economically and curricularly expedient. MIS programs eschew the core technology and theory cells of computer science and the practice cells of design / development. They avoid the infrastructure costs of the computing environment and the faculty with technical expertise needed to teach these. Somewhat ironically, they often prominently profess their commitment to "information technology" in their advertising and recruitment.

Shifting emphasis once again, this time closer to the core discipline of computer science, we arrive at the information systems discipline (sometimes called computer information systems). (See figure 5.) There is typically a premium placed on the practice aspects of IT and business. The treatment of computer science and engineering concepts are channeled toward effective decision making in application development. The treatment of business concepts contribute to the professional understanding of the business domain within which the application systems will be judged successful or not.

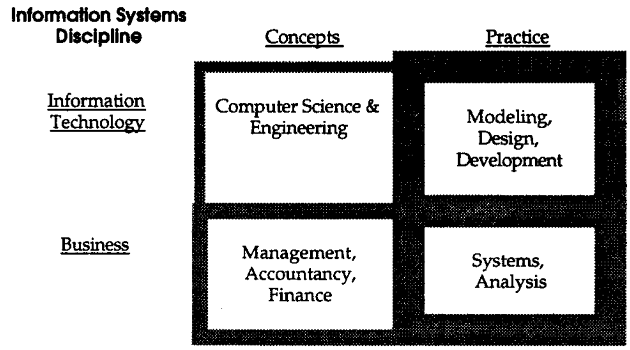


Figure 5 - Information Systems Emphasis

IS and CIS programs represent substantial institutional commitments to the preparation of profession-ready graduates. The investment in teaching resources which effectively support both the business context and the practical application of technology to business has been and will continue to be an economic and logistical challenge. It is just this mix of emphasis in academic and professional education that is most frequently sought after by IS recruiters of Fortune 500 companies[4]. And, the importance of this mix is clearly reflected in the curriculum guidelines of IS '97.

IS curricula are dependent on learning in both the business and IT disciplines. The need for learning in both disciplines usually results in IS programs constructed as parallel learning tracks. The tracks are normally sequenced independently to best serve the pedagogical goals of that track. Inter dependencies (prerequisite structures) may not adequately serve the needs of students who must learn from both tracks. The different tracks generally must serve first the needs of their primary customers. IS students are not usually the primary customers of business courses.

Accrediting bodies do not simplify the problem of accommodating different sequences of content learning. They often designate the treatment of business material at certain levels of student maturity (frequently in the last two

[4] Personal Correspondence, Information Technology Advisory Board, Bentley College, March, 1997.

academic years). IS curricula which attempt to serve all these priorities equally would probably result in fourth year student taking 20 or so courses?!

Herein lies a challenge for the IS or CIS curriculum designer. How can students be equipped with application domain knowledge "just in time" for the effective acquisition of systems, analysis, modeling, and development skills and concepts? And how can this be done when the sequencing of learning units in business and information technology may be independent of one another?

Live Case Study

Experiential learning (also called "learning-by-doing") is the most effective pedagogy in preparing IS professionals. Experiential learning of advanced IS methodologies and skills is a real challenge for student and teacher. The primary difficulty lies in the fact that "industrial strength" professional practice works best on "industrial strength" problems. Professional practice experiences with C.A.S.E. tools, project management, problem decomposition techniques, data modeling methods and tools, version control systems, documentation management disciplines, group ware systems, etc. – none of these "scale down" effectively. Using these professional methods or tools on undersized problems leaves students (and sometimes teachers) confused and often frustrated. The details of the methods combine with the intricacies of the application problem and domain. Students often fail in their attempts to grasp professional practice not because of their ineptness with the technology or concepts, but rather because of an ignorance of the essence of the problem and its domain.

A typical approach to providing students with practical experience in professional practice is the live case study. A live case study is an exercise in which the students participate (e.g. analysis, design or development, etc.) designed to illuminate particular theories or issues of professional IS practice. The task may be individual or team oriented. From the teacher's perspective the more authentic the experience the more expensive the task of preparing the student to understand the application problem itself, let alone the technology to be applied. The aspect of the problem that will be most directly manipulated, that part which is directly in contact with the technology being applied is usually not particularly complex. Rather, it is the contextual motivation found in the application domain for seeking a particular result, for applying the technology in the first place that is lost without the student's understanding of the problem's domain. Herein lies the teacher's dilemma. Does the teacher make a large time investment (in and out of the class) to prepare a realistic application domain for the student to understand and then spend the time to teach the student about the domain? Time spent on the application domain itself is not spent on the technology application experience. Or does the teacher prepare a more simplistic experience of applying the technology which may not convey the professional perspective that the student needs to have? If the technology experience does not engender confidence in the students, the students are unlikely to develop confidence in the technology or in their own use of it. Both of which are primary goals in the development of any professional! (Technology transfer research shows that methods and tools in which professionals have little faith are seldom, if ever, applied. And when they are applied they are done so halfheartedly[5].)

Threaded Live Case Study

One possible pedagogical vehicle for providing "just in time" application domain knowledge is the threaded live case study . A threaded live case study defines an application domain[6] enclosing several application problems. It is designed to inform the student about the context surrounding the application problems. Each problem is a platform upon which specific IS methodologies, technologies or analyses may be applied to achieve specific learning objectives. Working each problem imparts knowledge and experience with particular principles, concepts, and skills of professional practice. A threaded case is intended to be incrementally presented to the student. Each increment unfolds new aspects of the application domain which in turn present a new opportunity to explore IS practice. Experiencing multiple aspects of professional practice within the same application domain allows the students to explore the interplay that exists among theories, methodologies, and tools – reflecting a system of professional behavior rather than collection of loosely related technical ideas.

How can threaded live case studies be integrated into curricula? The application domain embodied in a threaded case can be conveyed to the students in increments over a period of time – over a variety of courses. In this manner the overhead of teaching the domain is amortized over the number of problems it enfolds or that can be derived within it. The students' domain knowledge is reinforced and expanded with each problem application.

The faculty sharing the threaded case study also share collateral benefits. Each can focus on their respective area of interest relieved (to varying degrees) of having to teach the whole application domain themselves. This savings in time and effort may be applied to the technology focus of the application problem.

---

[5] Raghavan, S.A. & Chand, D.R., "Diffusion of Software Engineering Technology," IEEE Software, July 1989, pp. 81-92.

[6] Waguespack, L.J., "Domain Analysis is an Essential Skill of the OO Analyst," Proceedings of ISECON '94, October 28-30, 1994, Louisville, KY, pp. 186-193.

The expected benefits of threaded live case studies may be seen in three basic ways: First, students receive an enriched set of individual application problem experiences. The experiences are richer because their understanding of the enclosing application domain is more complete than could be otherwise expected. Second, students develop a better overall understanding of the professional practice. This is because solving several problems (even with different technologies) within the same domain makes the interrelationships within the professional practice more easily observable. Third, the natural integrating effect on the courses threaded by the case study strengthens the students' perception of the cohesiveness of the curriculum. A sense of "wholeness" about the professional practice makes students more confident in the technology and in their overall grasp of their discipline.

Applying Threaded Live Cases in BSCIS'96

The BSCIS'96 curriculum was developed at Bentley College to answer a variety of professional and institutional challenges [Waguespack et al. ibid]. As an AACSB accredited business program the BSCIS'96 includes the following CIS course requirements:

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

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

Here is one illustration of a threaded live case study, Pat's Checkbook Assistant; the overview is found in Appendix A. It is conceptually simple, but rich with details. Let's take Pat's case and demonstrate how this domain may be threaded through the BSCIS curriculum as outlined above.

Pat's Checkbook Assistant: this case appears on the surface to be a simple problem of record keeping and reporting. Experience with this case shows that few undergraduate students and some graduate students are unaware of the basic process of checkbook accounting and account reconciliation. The case provides the following example application problems in Pat's domain applied to the courses indicated below.

CS210: Apply Pat's requirement to existing off the shelf personal computer products and evaluate the ease of use, breadth of function, and relative cost of the products as an exercise in informed consumer practice in the software market.

CS230: Apply Pat's requirements to the calculations required to determine current balance, and periodic reconciliation of checking entries based on monthly bank statements.

CS231: Prototype various user interfaces for the collection and manipulation of checking and savings account information including the format of account status and balance screens.

CS232: Develop an algorithm for the data entry and maintenance of the check register including the categorization of income and expenses for simple reporting of account status.

CS340: Assuming Pat's interface with the institution housing the accounts is electronic rather than postal, what PC system resources are required to connect to the bank? What resources are required at the bank's site to support multiple customers and possible online transaction transfer?

CS360: Model Pat's requirement using contemporary analysis and design methodologies and tools: DFD's, structure chart, Warnier diagrams, ERD's. Design the data model to maintain Pat's information and the transaction protocols to support data entry, reconciliations and tax-related information reporting.

CS361: Develop a relational database to support Pat's information using SQL. Define the necessary transaction protocols including integrity constraints to support error detection and data recovery contingencies. Develop a suite of SQL queries to generate all the nominal reports that Pat needs for monthly and annual budget management functions.

CS362: Document Pat's information and processing requirements in a contemporary C.A.S.E. tool (e.g. Visible Analyst). Generate the appropriate data description materials to create an SQL based database for Pat's data.

CS440: Evaluate and design a banking system that utilizes Pat's Checkbook Assistant as a client application for distributed information processing. Expand the scope of services provided to Pat to include loan processing, securities marketing and portfolio management. Identify the necessary "partners" to provide market information and their communication requirements.

CS450: Model Pat's requirements in object oriented analysis using appropriate Coad/Yourdan or Rational Rose diagraming disciplines. Prototype the application using Smalltalk, C++, PowerBuilder, or Delphi.

CS460: Plan and implement a consumer product that satisfies Pat's checkbook requirements. Include the necessary expansion of Pat's requirements to meet a reasonable market segment. Generate all the necessary project plans, documents, prototypes, test suites and system management documentation required to deliver the product to a public market.

This illustration of Pat's Checkbook threaded case can be easily expanded to include more problems per course or variable problems per course if there is concern that student information passed down from semester to semester is a problem. In each of the problems above additional documentation of Pat's domain would be provided to support the specific professional practice experience indicated. Pat's Checkbook threaded case is a good example of a conceptually simple problem domain which yields many rich learning opportunities. Many other rich domains are readily approachable.

Implications for Integrated Curriculum Development

Maintaining a relevant, current, and well integrated curriculum is a difficult task. This is true both within an academic department or across departments in a school of business. The threaded live case study vehicle is a practical means of fostering cross-fertilization of ideas, problems, and curricular concerns among faculty within an IS department. The vitality of a curriculum and the resulting quality received by the students depends to a large extent on the awareness each faculty member has of the entire program. The process of convening and working together to develop threaded case study material within a department can be a generally painless means of creating an exchange of information among faculty who may not often teach outside their specific specialty.

Group development of threaded case study material should be a relatively risk-free exercise of faculty interaction. This is because each participant is seeking to contribute to the domain description in order to support their own specific application problem support. In this way the domain definition results in almost no "competition." The usual tradeoffs that must be battled over for "real-estate" in the catalog or courses in the degree program are absent because a domain always has room for more ideas and content. It is probably not necessary to "dump" something in order to make "room" for something else.

The range of potential impacts of the threaded case study may be broad indeed. The low risk nature of threaded case study development may also be a useful vehicle for developing channels of communication between academic disciplines. Particularly in the case of Pat's Checkbook case one can see ample opportunities to explore Pat's domain in basic accountancy and finance as the specifics of business practice in those domains become important to evolving Pat's domain. If a threaded case study could be developed across an entire business curriculum that might be a powerful institutional marketing opportunity. Institutional development might also benefit as business organizations were solicited not only for domain information for cases but, also for support using their cases in curricula. One might envision a clearing house where threaded case studies are submitted by programs around the country and distributed for use wherever beneficial.

Summary

In this paper I have discussed disciplinary and resource tradeoffs that must be normally addressed in offering IS education in a business college environment. Those tradeoffs limit the options available in delivering high quality practical experience to IS students. The threaded live case study concept presents an intriguing pedagogical vehicle to overcome those limitations. The threaded case study is a simple concept. But, the benefits to students, faculty, curricula, departments and schools are promising enough to merit a degree of consideration in your environment as it has in ours. The example case and the application problems applied to the BSCIS program at Bentley College may serve as a framework for developing threaded live case studies in other IS programs.

Appendix A
Pat's Checkbook Assistant Case

An every day citizen, Pat, tries to keep track of his/her financial accounts with check register and written notes, but feels that a "personal financial assistant" would be very useful.

Pat maintains a checking and savings account. Pat writes checks and records them in the check register. Pat transfers money back and forth between the checking and savings accounts. At the end of the month when the bank statement arrives Pat tries to double check it by cross checking the bank statement with the checkbook's check register.

Pat tries to keep track of expenses by summarizing those expenses at the end of the month into groups for things like food, rent, car payments, entertainment, etc.

At the end of the year Pat's tax return usually requires itemized deductions and the checkbook is the primary source for that information.

The Global Information Systems Professional: Some Preliminary Observations For Education

Sanjeev Phukan
Professor of Business Administration
Bemidji State University
Bemidji, MN 56601

Abstract

The Global Economy has arrived. As individuals, we increasingly feel its impact on our day to day lives, from the things we buy to the places we travel. Even the smallest of businesses are often affected by the global marketplace. In many instances, the profitability, and even the survival of organizations depend on their ability to successfully compete in this global marketplace. Information Systems and Technology have played key parts in the creation of this global economy, and it takes no crystal ball to see that they are likely to play an ever increasing and important role in its future. Organizations that can create, maintain, and manage effective global Information Systems will likely achieve significant competitive advantages over those that cannot. Obviously, the management of these global organizations will require skills significantly different from those required to manage solely domestic organizations. This is equally true for Information Systems (IS) professionals that will operate in this global arena.

GLOBALIZATION AND THE WORLD ECONOMY

The 1980's finally saw the globalization of business in all its glory. Today, many of the old political and trade barriers between countries have crumbled and fallen. Although political boundaries between countries still exist, on a map that shows the real flows of financial and industrial activity, those boundaries have largely disappeared. (27). Ties between nations and regions are tied more to business and economic expediencies rather than any alignment of political ideology. For instance, it is widely rumored that the real reason that the Berlin wall came crashing down was because the East Germans wanted to go shopping in a big way! The transnational corporation has become commonplace, as companies use Information Systems (IS) and Information Technology (IT) to connect and manage enterprises that were often scattered all over the globe. The virtual corporation - a corporation that possesses no assets in the traditional sense - is increasingly making its presence felt as companies transport goods and services from all over the world to their customers, who are also all over the world.

An increasing level of worldwide economic activity today depends on the operations of these organizations such as these (14). Changes and advances in information technology, including computers, telecommunications, and office automation have brought global markets and global competition to the doorstep of even the smallest companies (4, 26). No major industry has escaped the effects of this globalization. We buy clothes, automobiles, and lumber from overseas markets. Our grocery stores provide us with food from distant places. We bank at institutions that have a worldwide presence. Our homes are filled with products that have been manufactured in dozens of different countries. Nothing is really local anymore, and to many businesses, any place is just the same as any other place. As Jim Hoagland, and American newspaper columnist wrote, "anywhere is rapidly becoming everywhere."

There have been many responses to this globalization. Nations and regions have formed strategic economic alliances such as the European Union and the North American Free Trade Agreement. Many businesses have had to reevaluate the way they operate to be able to successfully compete in a worldwide market. The skills that an individual must possess to perform effectively in this new international arena are also being redefined, and institutions of learning are changing their curricula so that they may provide these skills.

In the United States, as in other countries, it is essential that firms develop a global outlook. Many industries that were once mainstays of the US economy have either been entirely lost to foreign competition or have been overshadowed because of their inability to compete effectively in the global marketplace. Globalization will be a continuing and even increasing trend (26). The information technology that has driven this globalization has created an entirely different kind of economy - an information economy that is as different from a manufacturing economy as the manufacturing economy was different from the agricultural economy that prevailed at the turn of the century.

The repercussions for business will include both opportunities as well as pitfalls and barriers. Companies that are willing to seize the opportunities by being adequately prepared, by optimizing efficiency, by responding quickly to changes in the global marketplace, by continually learning, stand a greater chance of success in this marketplace (1, 30, 31). On the other hand, firms that do not adapt as well will find themselves ill-equipped to meet the challenges of rapidly changing markets and foreign competition. At the very least, these firms will probably suffer losses in both market share and profitability.

To survive in today's complex and volatile marketplace, businesses must become information based. (12). The use of Information Systems (IS) and Information Technology (IT) to enhance a firm's competitive advantage, to change and streamline its organization structure, to bring efficiency to its operations, to cope with increased levels of complexity and uncertainty, has been researched and recognized (4, 6, 26, 30, 33). IT and IS are the weapons of choice in this new battle for survival, and the caliber of a firm's IS will play a major contributing role in determining its continuing overall success. In the global marketplace, where the transnational, multinational, or virtual corporation operates, an additional requirement becomes evident: the ability to compete effectively in a global economy will depend not only on the effectiveness of the firm's IS, but also upon the global orientation of this IS (2, 18, 20). Since these organizations operate across products, markets, nations and cultures, they face problems and situations that are often far more diverse and complex than those faced by even the largest domestic firms (14). It follows that the management of the global IS of a transnational, multinational or virtual organization is also usually more complex and more difficult than the management of an IS that operates strictly within a domestic market. Nevertheless, to these organizations, the potential benefits of an effective global IS are enormous, as IT can provide tremendous advantages in overcoming barriers such as time zone differences, geographic separation and training difficulties. There is little doubt, therefore, that the issues relating to the management of global IS and IT have become increasingly important to most organizations that have a global presence. As technology shrinks the world, fewer and fewer businesses will be able to escape the touch of the global economy in some form or other, and even smaller companies will find that a global IS is essential for growth and survival (26).

IT allows even the smallest firms to become global. The implications in terms of larger markets, growth, and profitability are too significant to ignore. Effective use of IT in the global marketplace can allow smaller entities, whether they be companies or countries, to successfully compete with their larger counterparts. An excellent example of this opportunity effectively exploited is afforded by the tiny island nation of Singapore. With no natural resources to speak of, Singapore has managed to take its place in the world as a developed, newly industrialized, and technologically advanced nation. Much of this change has taken place in the last thirty years. There is little doubt that one of the prime movers in this transformation is and has been the national commitment to IT (17). Singapore's technology infrastructure is one of the most advanced in the region and compares favorably with much of the rest of the world. The results of this investment in technology speak for themselves. Singapore enjoys an excellent standard of living and an impressive per capita GDP. The port of Singapore moves more cargo than any other port in the world, thanks in part to its technology intensive TRADENET system which expedites trade documentation (21). Foreign investment in Singapore has grown in leaps and bounds, as global organizations exploit the potential offered by a stable government, skilled professionals, and state of the art technology and telecommunications.

Although the potential rewards of an effectively managed global organization are many, it is not all smooth sailing. Many issues that are unfamiliar or unknown often confront the global manager, including the global IS manager.

THE GLOBAL IS PROFESSIONAL

In today's global economy, IS professionals have to deal with a wide range of problems that may range from technical telecommunications issues to global corporate management (9). The management of IS on a global basis poses special challenges for IS managers (10) and the global IS professional will need a far greater depth and breadth of knowledge in comparison to his or her domestic counterpart (5). Obviously, it follows that the successful global IS professional will need to possess a sound grasp of IS standards and practices that exist in areas of the world other than their own.

In his discussion of the Multinational CIO, Roche (33) suggests that the CIO must help build the infrastructure that will meet present short term needs, but will also possess the flexibility to meet the longer-term needs of the global organization. The CIO will need to balance knowledge about information technology against the need to communicate with line level and top management.

One of the most difficult tasks that the multinational CIO must face is the simultaneous management of several learning curves: the IS organization, the different national organizations, the corporation as a whole, and the country management. This requires consummate political skill.

The task of building an infrastructure that will allow the corporation to change rapidly and adapt to new situations without severe damage is one that falls increasingly on the CIO's shoulders.

In their discussion of the roles of the Chief Information Officer (CIO) and the Global Information Officer (GIO), Kanter and Kesner (19) state that the GIO must possess capabilities above and beyond the attributes considered necessary for a successful CIO. The CIO of any operation will need to have a broad understanding of IT in all its aspects. The authors identify six critical factors that they believe must be present in order for the GIO to succeed. A brief examination of these factors is useful.

- Management Style and Leadership. Qualities of style and leadership essential for success include a strategic focus, flexibility in handling tactical issues, a people and task oriented project management style, an ability to delegate, consensus building, and a team approach. An understanding of people and cultures is vital.

- Organization and Structure of the IT Function. The GIO and the IT organization must be appropriately positioned within the overall organization, with the GIO reporting to a senior executive officer, preferably the CEO.

- Skill Base. The GIO must have a broad understanding of IS technologies, a comprehensive knowledge of the corporation, and an understanding of the countries and cultures in which the corporation operates. Since no one person will necessarily possess every attribute that is desirable, the GIO must build a team that possesses the necessary strengths.

- Total Quality Project Management. The GIO and his/her team must be committed to the concept of total quality.

- The Environment. The GIO should treat the environment as a pool of opportunities rather than one of obstacles. The establishment of alliances outside the organization in countries other than the organization's 'home' country is likely to prove beneficial in the long run.

- Technology Transfer and Change Implementation. The GIO should encourage the infusion of new hardware, software and applications into the IT group, as well as be a pioneer in the creation of a corporate culture that is responsive to change.

Other authors appear to reach similar conclusions. McFarlan (24) points out that much of the research that has been carried out concerning the nature of the CIO's job has assumed an almost totally domestic orientation, whereas the reality is quite different. He points out that the CIO that is responsible for international operations must contend with all the domestic issues as well as many additional problems. Factors such as the geographic transfer of work, global networking, global service levels, time based competition and cost cutting have all contributed to the increasing importance of global IS, and consequently to the increased reliance on global IS personnel. These global IS professionals must deal with a variety of additional transnational problems, including those of inadequate infrastructure, sketchy technical support, language problems, geographic separation, differing time zones, and greatly different labor technology cost tradeoffs.

It is easily concluded from this discussion that the tasks faced by a global IS professional are extraordinarily challenging and complex, requiring special skills and training. It would not be reasonable to assume that a successful 'domestic' IS professional would automatically be successful in a global sphere of operations, although he or she would certainly have a head start.

IMPLICATIONS FOR EDUCATION

How then should we prepare IS professionals for this global agenda? The preparation of IS managers and professionals for a global career is a challenge, the resolution of which will require cooperation between IS practitioners, corporate executives, and academics (26). An appropriate starting point is an examination of IS education. The heyday of a business school education appears to be over. The 1970's and 1980's were days of plenty for the nation's business schools, a time where they could ignore important trends with impunity (16). Those days of complacency are over and have been replaced by times characterized by rapidly declining business school enrollments. It is increasingly important that business and IS curricula in the 1990's be relevant, and schools can no longer afford to ignore the demands placed on them by a rapidly changing global economy.

Academics have long acknowledged the need for the internationalization of business school curricula. A study conducted by Nehrt (25) concluded that a majority of AACSB member schools intend to bring about some degree of internationalization in both their undergraduate and graduate business curricula in response to changes in AACSB accreditation guidelines. Other studies, notably the one by Thanapolous and Vernon (35) lend support to this conclusion. Both studies, however, unfortunately also concluded that the proportion of schools that actually implemented this decision was much smaller than would be considered desirable. The study by Porter and McKibbin (29) confirms this rather bleak state of affairs. The studies also concluded that business faculty lacked a proper understanding of international issues. Dunning (13) makes a strong case for an interdisciplinary approach to the study of International Business (IB) points out that many academic organizations have been active in seeking the internationalization of a variety of academic curricula. Deans and Ricks (11) support the interdisciplinary argument by suggesting a research framework that provides an interface region for IS, IB and other functional areas.

Several authors have studied the present status of global IT education and have suggested methods by which a global perspective can be incorporated into IS curricula. In his study, Palvia (27) reaches a more optimistic conclusion than the studies mentioned previously, and suggests that there is a growing trend towards global IT education in business schools in the United States. He does conclude, however, that initially most schools will offer a single course in International IS. He also concludes that the number of books in the field is limited and that educators will need to undergo formalized training in the area of International IS.

Deans and Goslar (7) concur in that they find very few resources currently available that address the international dimensions of IS. They further conclude that most IT curricula are at present devoid of a well-defined international component. The most recent curriculum guidelines from the Association of Computing Machinery (ACM) and the Data Processing Management Association (DPMA) contain no efforts to include an international component, even as a recommended elective. Deans and Goslar (7) offer two approaches to internationalization of the IS curriculum - an infusion approach, which incorporates an international element throughout all IT courses, and a freestanding approach, which creates a separate international IT course. The former approach allows for the diffusion of the international element to the greatest number of students, while the latter provides a more in-depth understanding to fewer students. Eom (15) also provides suggestions for an infusion approach to incorporate international elements into introductory and advanced MIS courses.

Loch and Khosrowpour (22) identify six key elements necessary to effectively internationalize the IS curriculum. These are (a) a strong relationship with business, to enhance relevancy, (b) an affiliation with other academic institutions, rather than an adversarial relationship, (c) faculty with international experience and an understanding of international issues, (d) the incorporation of international teaching materials, (e) hands-on work experience in another culture for students, and (f) language studies. The authors maintain that these are all actionable strategies. The authors also point out that in the United States several obstacles exist that hinder the development of an international IT curriculum. There appears to be no agreement as to what the international dimension should include and how it should be incorporated into existing curricula. IS is often recognized as a subset of another discipline rather a discipline in its own right. This makes it difficult to change the curriculum. Finally, many journals have yet to recognize international issues as an important research topic.

CONCLUSION

Based on the literature, it would be reasonable to conclude the following: First, the globalization of business is an established fact in today's world. This globalization will increase over time and an increasingly larger number of firms will participate actively in the global economy. Second, the currency for success in this global arena appears to be the effective use of IT. Global organizations that manage their information resources effectively will gain a competitive edge over those that do not. Third, the management of IT on a global scale involves opportunities, pitfalls, and issues that extend beyond those encountered within a purely domestic operation in both scope and complexity. Fourth, the IS professional that will be able to effectively manage IS and IT on a global scale must possess skills and attributes above and beyond those required by his or her domestic counterpart. Fifth, the development of such an IS professional will require the concerted efforts of educators, executives, and IS professionals. Finally, assuming that education is the first step in this process, available resources, although growing, are less than what may be required. Empirical research into the issues has begun, but has not reached nearly the level that has been directed towards 'domestic' issues. Business schools have yet to internationalize their curricula to any major extent. Overall, it would appear that both research and practice need to grow in this rapidly growing and critically important area.

REFERENCES

1. Bartlett, C.A. and S. Ghosal. "Managing Across Borders: New Organizational Responses," Sloan Management Review, 29:1, 1987, pp. 43-53.

2. Benjamin, R.I., J.F. Rockart, M.S. Scott Morton and J. Wyaman. "Information Technology: A Strategic Opportunity," Sloan Management Review, Spring 1984, pp. 1-10.

3. Buss, M.D.J. "Legislative Threat to Transborder Data Flow," Harvard Business Review, May-June 1984, pp. 11-18.

4. Cash, J.I. , F.W. McFarlan, J.L. McKenney and L. M. Applegate. Corporate Information Systems Management, Third Edition, Irwin, Homewood, IL, 1992, pp. 88-569.

5. Cheney, P.H. and G.M. Kasper "Responding to World Competition: Developing the Global IS Professional," Journal of Global Information Management, 1:1, Winter 1993, pp. 21- 31.

6. Davis, G.R. "MBA Students Need More MIS,", Datamation, September 1, 1986, p.19.

7. Deans, P.C. and M.D. Goslar. "Alternative Curriculum Approaches for Global IT Education Settings," in M. Khosrowpour and K.D. Loch (eds.). Global Information Technology Education. Harrisburg, PA: Idea Group Publishing, 1993, pp. 53-79.

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

9. Deans, P.C., K.R. Karwan, M.D. Goslar, D.A. Ricks and B. Toyne. "Identification of Key International Information Systems Issues in U.S.-Based Multinational Corporations," Journal of Management Information Systems, 7:4, Spring 1991, pp. 27-50.

10. Deans, P.C. and D.A. Ricks. "MIS Research: A Model for Incorporating the International Dimension," The Journal of High Technology Management Research, 2:1, 1991, pp. 57-81.

11. Deans, P.C. and D.A. Ricks. "An Agenda for Research Linking Information Systems and International Business: Theory, Methodology and Application," Journal of Global Information Management, 1:1, Winter 1993, pp. 6-19.

12. Drucker, P.F. "The Coming of the New Organization," Harvard Business Review, 66:1, January-February 1988, pp. 45-53.

13. Dunning, J.H. "The Study of International Business: A Plea For a More Interdisciplinary Approach," Journal of International Business Studies, Fall 1989, pp. 411-436.

14. Egelhoff, W.G. "Information Processing Theory and the Multinational Enterprise," Journal of International Business Studies, 22:3, Fall 1991, pp. 341-368

15. Eom, H.B. "Integrating International Dimensions Into MIS Courses: A New Challenge to Information Systems Educators," The Journal of Computer Information Systems, Spring 1990, pp. 11-14.

16. Evangelauf, J. "Success Has Spoiled Business Schools, New Study Charges," The Chronicle of Higher Education, 34:31, April 13, 1988, pp. A1, A20-21.

17. Gable, G.G and K.S. Raman. "Government Initiatives for IT Adoption in Small Business: Experiences of the Singapore Small Enterprise Computerization Program," International Information Systems, 1:1, January 1992, pp. 68-93.

18. Ives, B. and S.L. Jarvenpaa. "Applications of Global Information Technology: Key Issues for Management," MIS Quarterly, 15:1, March 1991, pp. 33-49.

19. Kanter, J. and R. Kesner. "The CIO/GIO as Catalyst and Facilitator: Building the Information Utility to Meet Global Challenges," in S. Palvia, P. Palvia and R.M. Zigli (eds.). The Global Issues of Information Technology Management. Harrisburg, PA: Idea Group Publishing, 1992, pp. 465-483.

20. Karimi, J. and B.R. Konsynski. "Globalization and Information Management," Journal of Management Information Systems, 7:4, Spring 1991, pp. 7-26.

21. King, J. and B. Konsynski. "Singapore Tradenet: A Tale of One City," in J.I. Cash, F.W. McFarlan, J.L. McKenney and L .M. Applegate. Corporate Information Systems Management, Homewood, IL, 1992, pp. 230-246.

22. Loch, K. and M. Khosrowpour. "Information Technology Education in Global Settings," in M. Khosrowpour and K.D. Loch (eds.). Global Information Technology Education. Harrisburg, PA: Idea Group Publishing, 1993, pp. 3-29.

23. Mannheim, M. "Global Information Technology: Issues and Strategic Opportunities," International Information Systems, 1:1, January 1992, p.38-67.

24. McFarlan, F.W. "Multinational CIO Challenge for the 1990s," in S. Palvia, P. Palvia and R.M. Zigli (eds.). The Global Issues of Information Technology Management. Harrisburg, PA: Idea Group Publishing, 1992, pp. 484-493.

25. Nehrt, L.C. "The Internationalization of the Curriculum," Journal of International Business Studies, 18:1, Spring 1987, pp. 83- 90.

26. Neumann, S. "Issues and Opportunities in International Information Systems," International Information Systems, 1:4, October 1992, pp. 1- 13.

27. Ohmae, K. "Managing in a Borderless World," Harvard Business Review, May-June 1989..

28. Palvia, P. "Assessment of Global IT Education in the United States," in M. Khosrowpour and K.D. Loch (eds.). Global Information Technology Education. Harrisburg, PA: Idea Group Publishing, 1993, pp. 156-180.

29. Palvia, S. and S. Saraswat. "Information Technology and the Transnational Corporation: The Emerging Multinational Issues," in S. Palvia, P. Palvia and R.M. Zigli (eds.). The Global Issues of Information Technology Management. Harrisburg, PA: Idea Group Publishing, 1992, pp. 554-574.

30. Porter, L.W. and L.E. McKibbin. Management Education and Development: Drift or Thrust into the 21st Century? New York: McGraw Hill, 1988.

31. Porter, M.E. and V.E. Millar. "How Information Gives You Competitive Advantage," Harvard Business Review, March-April 1979, pp. 137-145.

32. Porter, M.E. "The Competitive Advantage of Nations," Harvard Business Review, 90:2, March-April 1990, pp. 73-93.

33. Roche, E.M. Managing Information Technology in Multinational Corporations. New York: Macmillan, 1992.

34. Scott, J.P. and R.L. Hayen. "Global Information Systems: A Curriculum Challenge," Journal of Information Systems Education, 3:1, Spring 1991, pp. 14-16.

35. Simon, S.J. and V. Grover, "Strategic Use of Information Technology in International Business: A Framework for Information Technology Application," Journal of Global Information Management, 1:2, Spring 1993, pp. 29-42.

36. Singapore Facts and Pictures 1993. Singapore Ministry of Information and the Arts, Singapore, 1993.

37. Thanapoulos, J. and I.R. Vernon. "International Business Education in the AACSB Schools," Journal of International Business Studies, Spring 1987, pp. 91-98.

A Joint Double Degree Program Between an Australian and British University in Information Systems and Computer Science - a Venture into Informatics

Author

John A. Palmer
Senior Lecturer,
School of Information Systems
Curtin University of Technology
Perth, Western Australia.

Abstract

This paper describes the birth, a few years ago, of an idea of a truly International Information System education program, and how that idea foundered and re-surfaced. The concept being finally implemented involves two Universities in two countries of the world collaborating on a joint double degree program, where graduating students will obtain two degrees, one from each University. The amount of effort to launch such a program and the cooperation of the management of the two Universities are stumbling blocks that saw the demise of our first attempt to instigate such a joint double degree, but with renewed vigour and support from both Vice-Chancellors, the idea is rapidly approaching implementation.

Introduction

As early as 1993, Professor Marty Katz of Georgia College and the author collaborated on a staff exchange between his College and Curtin University of Technology in Western Australia. Working together for a semester on subject matter we both taught at our respective home institutions led us to contemplate the enormous potential of having students study at both venues, one in America, the other in Australia.

In a world of increasing competitiveness and globalisation of business, students would gain a considerable competitive edge by having studied Informatics in two countries, two cultures, two totally different and unique experiences. The advantages these graduates would have over their peers who had studied at only one University were at least two-fold. Firstly, they would have a greater appeal as potential employees because they would have spent at least two years in two separate and culturally different environments, and secondly, they would have commenced their life-long development of an International network of friends and associates.

Although the relationship with Georgia College and Curtin University for a number of reasons did not culminate in the implementation of a joint double degree program, sufficient groundwork had been done to enable another program with a British University to be developed. The joint double degree development between the University of Teesside, School of Computing and Mathematics, and Curtin University, School of Information Systems, is the subject of this paper.

The Background

Mr Steve Dunne, currently a Senior Lecturer at the University of Teesside had previously been on the faculty of the School of Information Systems, Curtin University. Mr Dunne had taught at Curtin for three years from 1987 to 1990, knew the Curtin program, the philosophy of the school, the syllabi of its courses, and had made several friends within the faculty. Over the course of time, Mr Dunne began to work closely with one of our senior academics, Mr Mike Newby, responsible for the review of our undergraduate degree course in Information Systems and Information Technology. Together they were instrumental in the reaccreditation of the degree and many of the enhancements to it. When it was obvious that the relationship with Georgia College was not leading us to the joint double degree program we had initially conceived, we put the idea to Mr Dunne who had by that time returned to Teesside.

Coincidently, Mr Newby was familiar with the University of Teesside as he had grown up in that part of the United Kingdom. It was, therefore, a rather natural choice of partner for Curtin when the Georgia connection did not eventuate. However, with Georgia, we were looking at a joint double degree of a Bachelor of General Studies from Georgia and a Bachelor of Commerce (Information Technology) from Curtin. Now, with the University of Teesside, School of Computing and Mathematics, we were no longer in the "general arts" area, but rather the discipline of Computer Science. How were we to marry a Commerce degree in Information Systems with a Computer Science degree? Fortunately, the University of Teesside had the answer in its flexibility of offering Science degrees in computing.

The Curtin Bachelor of Commerce (Information Technology)

The Curtin Bachelor of Commerce (Information Technology) course is a three year full-time degree program comprising four segments as follows:

- 8 general business subjects
- 12 subjects in the Information Technology major
- 4 subjects in any minor / or 4 elective subjetcs
- 1 elective subject

The University follows a two semester per year mode of teaching with an optional summer semester on a fee paying basis. The following diagram depicts the sequence of the general business subjects, major, minor and elective subjects over the six semesters.

	First Semester (Mar - Jun)	Second Semester (Aug-Nov)
1st Year	5 general business	2 general business 2 IT major
2nd Year	3 IT major 1 general business	3 IT major 1 elective
3rd Year	2 IT major 2 minor/electives	2 IT options 2 minor/electives

The joint double degree program assumes that students commence their studies in our second semester, the end of July. They will spend two years at Curtin University and then proceed to the University of Teesside for the start of their third year commencing late September. As it is unusual to start Curtin mid-way through the year, a special program of study was established to cater for these students.

It is a general guideline or University Policy that Curtin does not grant exemptions or advanced standing in final year subjects. If students were to depart at the end of second year they would miss two specific subjects from their IT major, and two optional subjects which must be in the IT area in addition to four minor or elective subjects. In order to avoid conflict with University Policy, we created a unique program of study to enable students to complete all mandatory subjects within two years.

Summer Semester	First Semester	Second Semester
		2 general business 2 IT major
2 general business	3 IT major 1 general business	3 IT major 1 bridging subject
2 general business	2 IT major 1 general business	

The above pattern enables the mandatory units normally taught in the third year to be moved into the second year by utilising the summer semester. As the optional IT subjects could, in fact, be second year subjects, we have maintained the integrity of the University's policy of not giving advanced standing to third year subjects. There are no mandatory third year subjects not included in the above pattern.

There may be a requirement to include an additional unit covering computer hardware and computer maths not normally covered in the Curtin degree but which Teesside would require for entry into their portion of the joint double degree. Again, by utilising the summer semester we can cater to the addition of this bridging subject. We can also allow for the repeat of any subject a student may have failed along the way as there are only three subjects taught in the final semester instead of the usual four subjects.

The above program is a bit heavier than normal as students will take an additional two subjects in their two years at Curtin. However, the program is still contained within a two year period. The timing is such that the students can migrate to the UK in time for a late September start and enjoy a vacation along the way.

The Teesside Bachelor of Science (Honours)

The University of Teesside Bachelor of Science (Honours) is a four year modular sandwich course involving three years of University teaching plus a one-year work-based supervised live project in industry in the Middlesbrough area which occurs in the students' third year of study.

The framework for the Modular Computing Scheme is depicted in the following table:

	First Semester	Second Semester
Year 1	5 designated modules	2 or 3 designated modules 3 or 2 elective modules
Year 2	3 or 4 designated modules 2 or 1 elective modules	4 or 5 designated modules 1 elective module
Year 3	supervised work experience in a year-long live project	
Year 4	2 or 3 designated double modules for whole year including a practical project 3 or 2 elective double modules for whole year	

In 1995 it became apparent that there was student demand from overseas, particularly France, for entry into a shortened version of the course on the basis of advanced standing through previous studies. Teesside re-engineered their course to allow students with prior qualifications to complete the Teesside BSc (Honours) in two years.

The course has several computing routes, some of which would be available to the joint double degree cohort. There are routes in Business Computing, Computer Science, Information Technology, Software Engineering, and Information Sciences, a catch-all route for students not wishing to follow any of the above specialities. The Information Sciences route may not lead to accreditation by the relevant computer bodies, in this case the British Computer Society, whereas the other routes are all fully accredited by the BCS.

The generic format of all the routes to the Bachelor of Science (Honours) accelerated two year program is as follows:

	First Semester (Sept-February)	Second Semester (March-June)
Third year	5 modules	work-based live project
Fourth year	2 or 3 designated double modules for whole year including a practical project 3 or 2 elective double modules for whole year	

The student's choice of modules depends on the specific route chosen as well as whether the student has done the prerequisite subjects either at Teesside or in previous studies at another University.

The Proposed Joint Double Degree Program

The net effect of combining the Curtin University Bachelor of Commerce program with the University of Teesside Bachelor of Science Honours program is a four year full-time course of study commencing in July and concluding in June. The awards offered to the students would be the award of Bachelor of Commerce (Information Technology) from Curtin University and the award of Bachelor of Science with Honours from the University of Teesside. The program also involves summer semesters at Curtin. The table below illustrates the sequence and country of study.

	Jan - Feb	March - June	Aug - Nov
Australia			first semester
Australia	summer semester	second semester	third semester
Australia	summer semester	fourth semester	
		March - June	Sept - Feb
England			fifth semester
England		sixth semester	seventh semester
England		eighth semester	

The program combines a number of educational elements which students will find incredibly useful in their quest for employment. The program approaches Kling's concept of Organisational Informatics (Kling, 1996), the marriage of the disciplines of Information Systems and Computer Science to create a field of study embracing the development of computerised information systems and communication systems in organisations.

Students will be graduates of one of the finest and most respected Business Schools and one of the oldest and best Information Technology degrees in Australia and South-East Asia. The School of Information Systems at Curtin University is the largest of its kind in Australia and one of the largest IS schools in the world. Additionally, students will also be science honours graduates from one of the highest ranking Universities in the United Kingdom.

The combination of skills in Business, Information Systems, Information Management, Computer Science, Software Engineering, which this course offers will be unique in an undergraduate program. The best of the "science" world is being combined with the best of "business computing" to offer graduates that competitive edge which so many students will find invaluable.

The Market

The Curtin Business School is one of the leading Australian Universities active in the South-East Asian market, both in attracting students from that region and in running programs at various twinning centres in the region. The Curtin Business School has had partners in Singapore, Malayasia and Hong Kong for over ten years and has had considerable success in marketing education in those countries. The University of Teesside has had several years experience in attracting students from Europe as well a small number from South-East Asia. Both Universities have the infrastructure to cope with growing number of International students.

Although the overseas fees in the two Universities are not identical, as Curtin has included extra tuition in their summer semesters, the variation in living costs between the two countries makes the costs paid in Australia comparable to the costs incurred whilst in Britian. For four years tuition fees, and four years living expenses students can acquire two very different degrees, from two very different cultures at about the same cost as acquiring an honours degree from either one of the Universities.

Marketing the program in countries from which significant numbers of students currently come, should be at least as successful as marketing any other education program overseas. If that is the case, Curtin University and the University of Teesside would expect a greater market share within the next few years than either institution currently enjoys.

Conclusion

What the program offers to overseas students is a unique opportunity to gain an internation business computing / computer science education, culminating in the award of two degrees from two differing Western cultures in four years. Furthermore, the program should cost the student little more than a normal four year honours degree in either Britain or Australia. With an increasing emphasis on International Business, understanding the business and computing environments in a variety of countries and cultures, this program offers a unique opportunity for students from both Australia and Britain, but particularly from South-East Asian countries such as China, India and Indonesia. They will experience diverse cultures and gain what no other program, to the knowledge of the author, can offer; the award of two Bachelor degrees in two countries within the time span of a single degree.

Appendix I

Details of the Curtin Bachelor of Commerce (Information Technology) program

<u>Semester 1</u>
5 general business subjects:
- Economics (Micro) 100
- Accounting 100
- Information Systems 100
- Law (Legal Framework) 100
- Management 100

<u>Semester 2</u>
2 general business subjects:
- Business Statistics 101
- Business Communications 101

2 Information Technology major subjects:
- Information Systems 102
- Program Design 102

<u>Semester 3</u>
3 Information Technology major subjects:
- Systems Development 201
- Systems Implementation 201
- Computer Environments 201

1 general business subject:
- Accounting (Managerial) 101

<u>Semester 4</u>
3 Information Technology major subjects:
- Systems Development 202
- Systems Implementation 202
- Database Systems 202

1 elective subject

<u>Semester 5</u>
2 Information Systems major subjects:
- Software Design 301
- Distributed Systems 301

2 minor or elective subjects

<u>Semester 6</u>
2 Information Technology optional subjects
2 minor or elective subjects

Appendix II

Details of the University of Teesside Bachelor of Science (Honours) program

Semester 1	Communications Computing Mathematics Principles of Computers 1 Principles of Software 1 Structured Systems Analysis
Semester 2	Principles of Software 2 Systems Design 3 elective modules
Semester 3	Management Information Systems 4 elective modules
Semester 4	Databases and 4GL's Networks & Communications 3 elective modules
Semester 5/6	Supervised year long work experience
Semester 7/8	Practical project 4 elective double modules

References

Kling, Rob (1996). *Computerization and Controversy Value Conflicts and Social Choices*, Academic Press, San Diego.

EDUCATIONAL COLLABORATION ON THE WEB: MARKUP INNOVATIONS FOR ADVANCED APPLICATIONS

Jeffrey Hsu
GARI Software/IDD Information Services
Rutgers University

Markup languages have gained prominence in light of the increased interest in electronic and Internet WWW publishing. These descriptive markup languages allow one to describe a text element or document in a way which is independent of its final output and form. One area which deserves greater attention in this regard is the creation of data elements for survey questionnaires and collaborative applications, and any comprehensive markup language standard should include markups for supporting these kinds of applications. The ability to create surveys and collaborative applications on the Internet has a wide range of educational possibilities, including conducting opinion surveys of all kinds, collaboratively judging the review of papers, and in promoting group decision judgments and decision making.

I. INTRODUCTION AND BACKGROUND

The power of markup languages is becoming recognized for the efficient formatting and processing of text, especially with the increased interest in electronic text manipulation, desktop publishing, the Internet, and the World Wide Web. Descriptive markup is of particular importance, because it allows a description of a document or text element which is independent of its final form and output. While many persons are familiar with the terms HTML or SGML, fewer may be familiar with the term "markup language." Markups are sets of tags, tokens, characters, or specialized commands (a collection of which comprise a markup language) which are placed into a body of text in order to provide information about the text or other data being processed. A markup could be as simple as a space or line feed, or a complex set of symbols for setting all the formatting details for a text document (Coombs et al., 1987; Goldfarb, 1981; Wright, 1992) Markups enable one to "unlock" the data content of a document. In other words, a document is not just a stream of characters, but rather is a data structure which encompasses a great deal of content (Cronk, 1993; Goldfarb, 1991; Van Herwignen et al. 1990). There are several different kinds of markups, the most important of which is the descriptive markup. Descriptive markup specifies what text element a unit of data is, and allows you to describe and classify it. A descriptive markup (command) language approach allows the creator of a document to define a number of "element types" or "data structures," which identifies a text portion as a member of a certain class. For instance, you can specify if a piece of text is a long quotation, table, paragraph, or a footnote (Coombs et al., 1987; Tuck, 1989; Blake, 1989; MacLeod, 1990; Van Herwignen et al., 1990). A practical example of an existing markup language standard is SGML (Standard Generalized Markup Language). This standard for marking up text has been defined by various publications (Association of American Publishers, 1992). A widely used subset of SGML is HTML (HyperText Markup Language), which is used to create hypertext-based documents on the Internet World Wide Web.

One of the most important benefits of a descriptive markup approach to survey design is its ability to minimize cognitive demands. Rather than recalling, selecting, and remembering codes for the creation of entry and coding of procedural markup, there is only one step involved in descriptive markup after recognizing the element type: to use the appropriate markup command together with the text to be "marked up." This frees the survey creator from formatting concerns inherent in traditional direct manipulation or WYSIWYG methods. The use of descriptive markup also has the benefits of better maintainability and portability. In terms of maintainability, in the case where the actual formatting or structure of a question (element) type needs to be modified, this can be done by changing the processing program or system, without affecting the markups or text. This allows the same data to be used across different applications and platforms, resulting in greater portability. (Coombs et al, 1987; Tuck, 1989; Blake, 1989; MacLeod, 1990)

Survey Applications: An important component of behavioral, social science, and information systems research involves the use of surveys. Questionnaires are an important method for conducting these surveys. Usually, a survey questionnaire would contain a variety of question types, such as Likert, semantic differential, free response, multiple choice, rank ordering, and others. Collaborative Applications: The collaborative markups proposed here are designed to support applications including group paper evaluation, group decision making, and related applications which involves categorization, estimations, and related types of evaluations. Markups are supported for rating papers, paper commentary, voting, future projections, discussion, and probabilities.

II. DESCRIPTION OF THE COLLABORATIVE MARKUP LANGUAGE (CML)

The Collaborative Markup Language (CML) proposed here is adapted from the Standard General Markup Language (SGML) concept, however it has been modified with respect to syntax to make it easier to input and use. The Markup Language Collaborative System (MLCS) is a software program designed exclusively for the creation of these specialized data elements. Its main goal is enable a user to create, edit, modify, and run online collaborative sessions using the markup language system. The program has the following features:

INTEGRATED DEVELOPMENT ENVIRONMENT. This is a complete integrated development environment for creating collaborative data elements.
GRAPHICAL USER INTERFACE. The entire system, including both the Custom Editor and the survey run module, offers a graphical user interface (GUI).
CUSTOMIZED EDITOR/LEARNING FACILITY. This allows for the creation, editing, and saving of the markup language text. A flexible, customizable help/learning facility system is featured.
PARSER (SURVEY GENERATOR) MODULE. This allows for the resulting online session to be presented in a graphical format. Error checking and related features are included.
HELP SYSTEM. There is a customized help system, not only for learning the system itself, but also for helping to create the markup language.
DATA COLLECTION MODULE. Records responses from respondents into a data file. The data collection is performed automatically and produces a file which contains all the responses.

DESCRIPTION OF MARKUP LANGUAGE ELEMENTS

RATING OF PAPERS

1. Explanation: Rate papers based on set criteria.

GENERAL FORM:

```
<rate_p>            // markup tag for paper rating
---- text ----      // text
<rates=n>           // criterion maximum limit
<list a/b/c/d/e....>  // rating criterion list
<paper=n>           // paper # maximum limit
<list 1/2/3/4/5....>  // paper listing identifiers
</rate_p>           // closing markup
```

EXAMPLE:

```
<rate_p>
Please rate the papers on a scale of 1 (highest) to 10
(lowest).
<rates=5>
<list clarity/originality/writing style/usefulness>
<paper=5>
<list 1/2/3/4/5....>
</rate_p>
```

2. Explanation: Allows for ranking of papers by different reviewers.

GENERAL FORM

```
<rank_p>              // opening markup tag
text of instruction string...
<reviewer=n>          //number of reviewers
<list r1/r2/r3/r4....>  //list of reviewers
<paper=n>             //number of papers
<list 1/2/3/4>        //list of paper identifiers
</rank_p>             //closing markup tag
```

EXAMPLE:

```
<rank_p>
Please rank these papers in terms of overall quality.
<reviewer=4>
<list Jeff/Jim/Bill/Frank....>
<paper=4>
<list 1/2/3/4>
</rank_p>
```

PAPER COMMENTARY

3. Paper Commentary: provides for entry of paper reference and comments by group members.

GENERAL FORM

```
<comment_p>                   //opening markup tag
text of instruction string    //text
<text1><text2>        //horizontal and vertical identifiers
</comment_p>                  //close markup
```

EXAMPLE:

```
<comment_p>
LIST YOUR COMMENTS HERE:
<page><comment>
</comment_p>
```

VOTING:

4. Vote: Enables voting for certain issues, ideas, or candidates; single or multiple selection.

GENERAL FORM:
<vote> //opening markup tag
text of instruction string //text
<single/multiple> //voting for one or more candidates
<candidate=n> // number of candidates
<list a/b/c/d....> //candidate names
<choice=n> //number of voters
<list w/x/y/z....> //voter identifiers
</vote> //closing tag

EXAMPLE:
<vote>
PRESIDENCY VOTE
<single/multiple>
<candidate=3>
<list Fred/Bill/Ted>
<choice=3>
<list A/B/C>
</vote>

CHANGE IN VARIABLE

5. Change in Variable. Examines the beliefs on whether a variable will change in the future (increase, decrease, stay the same).

GENERAL FORM

<change> //opening markup tag
text of instruction string
<variable=n> //number of variables
<list x1/x2/x3/x4...>//variables in list are specified here
<ch_factor=n> //number of change factors
<list y1/y2/y3/y4...> // change factor-descriptive var.
</change> //closing tag

EXAMPLE

<change>
INCREASE IN INTERNET USAGE
<variable=4>
<list V1/V2/V3/V4...>
<ch_factor=5>
<list SI/I/O/D/SD> // change factor -descriptive variable
</change>

FUTURE PROJECTION

6. Projection: projects the value of a variable into the future.

GENERAL FORM:
<project> //opening markup tag
text of instruction string //text
<variable=n> //number of variables
<list x1/x2/x3/x4...> //list of variables
<time_scale=n> //time change scales
<list y1/y2/y3/y4....> //time scale is normally //numeric (ex:in years to future)
</project> //closing tag

EXAMPLE:
<project>
What do you project as changes to these variables?
<variable=2>
<list crime/income>
<time_scale=5>
<list 1998/1999/2000/2001/2002> //time scale is //normally numeric (in years to future; year dates)
</project>

DISCUSSION

7. Discussion: Enables discussion, pro and con, on an issue or topic.

GENERAL FORM
<discuss> //opening markup tag
<text> //text markup tag
put the accompanying text here //instruction text
<statement> //statement for discussion
put the statement text here //statement text
<columns=n> // how many discussion columns?
Arguments + //first column
Arguments - //second column
Comments //third column
</discuss> //closing markup text

EXAMPLE:
<discuss>
<text>
Discussion Forum
<statement>
Do you like Network Computers?
<columns=3>
Arguments +
Arguments -
Comments
</discuss>

PROBABILITIES

9. Probabilities. Explanation: Entering of estimated or subjective probabilities for decision analysis.

GENERAL FORM

<prob> //opening markup text
<total/nototal> //total or nototal of probabilities
text /instruction string //text
<event=n> //total # of events
issue1 // 1^{st} issue
issue2 // 2^{nd} issue
issue3 //3^{rd} issue
</prob> //closing markup text

EXAMPLE:

<prob> //opening markup text
<total> //total or nototal of probabilities
What will we experience in the year 2000?
<event=3> //total # of events
nuclear attack // 1^{st} issue
peace // 2^{nd} issue
happiness //3^{rd} issue
</prob> //closing markup text

ALLOCATION OF RESOURCES

This type of question is designed to allow the user to allocate resources between various choices. The following is the general form of the allocation markup:

GENERAL FORM

<allocation>
--allocation question text--
<limit> <-- specifies allocation limit
<list /choice1/choice2/choice3/choice4/choice5....>
</allocation>

<allocation> is the question type identifier.

The allowed allocation limit comes next, This should be in brackets, such as <2000>.

The <list.....> specifies the choices.

This text :

EXAMPLE

<allocation>
ALLOCATION OF RESOURCES: You have a total of $5000.00
<5000>
<list /clothes/diamond/computer/boat>
</allocation>

will bring about the following output:
ALLOCATION OF RESOURCES: You have a total of $5000.00
Press the ESC key when done

clothes ?
diamond ?
computer ?
boat ?

Total ????

EXPLICIT RESPONSE

Frequently, it is desirable to get responses which are of a certain type or format. This kind of item asks for a specific response, such as 'Y' or 'N' or 'T' or 'F'.

GENERAL FORM

<explicit>
(question text)
</explicit>

EXAMPLE
<explicit>
Do you enjoy skiing?

FREE RESPONSE

Often, the need arises to allow the respondent to enter lines or paragraphs of text as a response.

GENERAL FORMAT

Each free response item or group of response items must have the following general markup:

<free response>
(text of the question)
</free response>

When this markup is executed, it will display the question, together with an editing screen which allows the user to create a free-form text response to the question.

EXAMPLE

<free response>
What are your thoughts about the future of the world?

LIKERT SCALE

Description

Frequently, the need arises to express various levels of agreement or disagreement with a statement or idea. The markup set up for this system allows for both the nominal (such as Strongly Agree...Strongly Disagree) and the interval scale type questions (example of 1..7 and the anchor points Strongly Agree...Strongly Disagree

GENERAL FORMAT

<likert> --required function name
<[nominal/interval]> --type specifier
--text of the question-- --question text
<endpoints /endpoint1 /endpoint2>
<list /choice1/choice2/choice3/choice4----->

There are four basic elements to the Likert question markup:
<likert> is the required function name; nominal/interval.

<endpoints/ ./ ./ .> will specify the particular endpoints which will be placed on each end of the scale.

<list /. /. /.> allows you to specify the individual choices which you are allowed to make in regards to the Likert question.

EXAMPLE
<likert>
<interval>
Summer is my favorite time of year.
SA=Strongly Agree, A=Agree, N=Neutral, D=Disagree, SD=Strongly Disagree
<endpoints /strongly agree /strongly disagree>
<list /SA/A/N/D/SD>

which will produce output appearing as follows:

Summer is my favorite time of year.
SA=Strongly Agree, A=Agree, N=Neutral, D=Disagree, SD=Strongly Disagree

: SA : A: N: D : SD :

Strongly Agree   Strongly Disagree

MULTIPLE CHOICE

Description

One of the most commonly used types of survey questions asks the respondent to choose between various options.

GENERAL FORM

Specification:
<multiple choice> --required function name
--text of the question--
<(alpha/numeric) (single/multiple) (horizontal/vertical)>
<list /choice1/choice2/choice3......>

The following are the options for this markup:

<multiple choice> is the required keyword identifier for this question type.

text of the question, without brackets, follows the keyword identifier.

The third set of commands specifies the multiple choice options, which includes:

a) alpha has alphabetic choices (a,b,c,d...).
b) numeric allows for choices using numbers
c) single means that only one choice can be chosen.
d) multiple means that more than one can be chosen .
e) horizontal format means that the choices will be listed side by side on the screen or on paper.
f) vertical format means that the choices are listed on separate lines.

<list / / /...> specifies the choices for each of the selections in the list. The individual choices should be separated by forward slashes.

EXAMPLE/ALPHABETIC.
<multiple choice>
What is your favorite season?
<alpha single vertical>
<list /spring/summer/fall/winter>
</multiple choice>

which will produce output as follows:

What is your favorite season?

a. spring
b. summer
c. fall
d. winter

Choice: []

RANK ORDER QUESTIONS

Explanation

Another important type of question concerns rank ordering of items, in terms of preference, necessity, or some other criterion.

GENERAL FORM

The general form of the markup is as follows:

<rank order>
--text of the question--
<list /choice1/choice2/choice3/choice4---->
</rank order>

<rank order> is the required keyword for rank order question markups.

the text of the question follows the questions, and is not placed in brackets.

<list /./.>/...> specifies the choices which you want to specify for the rank ordering.

EXAMPLE
<rank order>
Which food would you like best?
<list /pizza/ice cream/hot dog/apple pie>

</rank order>

Which food would you like best?

pizza { }
ice cream { }
hot dog { }
apple pie { }

SEMANTIC DIFFERENTIAL SCALE

GENERAL FORMAT
<semantic>
--question text--
<endpoints /endpoint1 /endpoint2>
<list /choice1/choice2/choice3/choice4/choice5...>
</semantic>

<semantic> is the required keyword for the semantic differential question text.

<endpoints /./..> specifies the two endpoint descriptions which are displayed on the screen for that question.

<list /././.....> specifies the individual choices which the respondent will select from when answering the survey.

EXAMPLE

<semantic>
Taking an exam is stressful.
1=Agree, 4=Neutral, 7=Disagree
<endpoints /agree /disagree>
<list /1/2/3/4/5/6/7>
</semantic>

which will produce the following output:

Taking an exam is stressful.
1=Agree 4=Neutral 7=Disagree

1	2	3	4	5	6	7
Agree					Disagree	

Conclusion

The use of descriptive markup in tagging the content of data elements within documents is a viable means for describing the content of a file of text, without becoming involved in the complicated specifics of applications, platforms, and different kinds of text processing systems and software. This paper proposes a set of markups which are designed as an extension to markup standards such as SGML which are designed to provide an easy-to-use and flexible means for creating collaborative data items. It allows the creation of these using a straightforward markup command language which minimizes cognitive demands, allows for more rapid development, and can easily be adapted for use on an electronic text-processing system. This could be used to add greatly needed collaborative and survey questionnaire data types to existing markup language standards.

References

Association of American Publishers (1992). Electronic Manuscript Project. Reference Manual on Electronic Manuscript Preparation and Markup.

Blake, M. (1989). Archiving of Electronic Documents. The Electronic Library, 7, 6, 376-385.

Coombs, J.H., Renear A.H., and Rose, S.J. (1987). Markup Systems and the Future of Scholarly Text Processing. Communications of the ACM, 30, 11, 933-947.

Cronk, R.D. (1993). Unlocking Data's Content. BYTE, 18, 10, 111-120.

Goldfarb, C. (1981). A Generalized Approach to Document Markup. In Proceedings of ACM SIGPLAN-SIGOA Symposium on Text Manipulation (Portland OR, June 9-10), ACM, New York, pp. 68-71.

Goldfarb, C. (1991). The SGML Handbook. New York: Oxford University Press.

Grudin, J. and Barnard, P. (1985). When Does an Abbreviation become a Word and Related Questions. Proceedings CHI '85, ACM, 121-126.

Landauer, T.K., Calotti, K.M. and Hartwell, S. (1983). Natural Command names and Initial Learning. Communications of the ACM, 26, 7, 495-503.

Macleod, I. (1990). Storage and Retrieval of Structured Documents. Information Processing and Management, 26, 2, 197-208.

Shneiderman, B. (1987). Designing the User Interface. Reading MA: Addison Wesley.

Tuck, B. (1989). Desktop Publishing: What It Is and What It Can Do For You. ASLIB Proceedings, 41, 1.

Van Herwignen, E. et al. (1990). Practical SGML. Norwell MA: Kluwer Academic Publishers.

Wright, S. et al. (1992). SGML Frees Information. BYTE, June 1992, p. 263.

Zadow, J. (1993). SGML, The Management Issues. New Media News, Fall/Winter 1993, pp. 28-33.

COMPUTER BASED TRAINING AND DISTANCE EDUCATION SUCCESS FACTORS AND EVALUATION TECHNIQUES

Thomas Janicki and Gregory Madey
Kent State University, School of Business Administration, Kent, OH 44242

ABSTRACT

New challenges emerge as the use of the computer and the internet to develop and deliver training modules to students and businesses increases. This article will review the current literature for the obstacles to consider and overcome in training module preparation, delivery and evaluation. It then synthesizes the key success factors in the development of computer based training. A key finding is the need for the training module to be constructed form a "learner centered" rather then a "teacher centered" perspective. The instructor is the facilitator of information, not the prime deliverer of that information. Successful evaluation techniques and other design concepts are also reviewed.

Introduction

The use of the computer and related telecommunication technologies to enhance the learning capabilities of individuals and organizations has grown exponentially over the past few years. This is a result of increased "distance learning" or "distributed learning" classes and training and learning modules developed using the computer as the primary learning resource. Both educational institutions and business organizations are finding the computer an excellent vehicle to deliver training and learning needs.

Utilizing the computer for learning applications does create new and different challenges for the developers and facilitators of this information versus traditional classroom training modules. This paper briefly discusses the different types of computer assisted learning. In addition we review the literature to identify and synthesize a framework of key success factors in the design and delivery of these new learning modules. Finally, we conclude with a discussion of the evaluation and measurement techniques to measure the effectiveness of these new learning tools.

Two broad areas of new technologies using the computer are discussed. The first is computer based training (CBT), and the second is distance learning and delivery applications to remote sites. Distance learning is also referred to as distributed learning or distance education. Overall, it is important that educational institutions embrace the capabilities of CBT and the electronic transfer of learning and teaching modules. Ives and Jarvenpaa [1996] assert that the private sector will eclipse the public sector as a predominant educational institution. They suggest that a goal of universities should be to work with industry to enhance their learning capabilities as a result of CBT and distance learning technologies. Many of the research articles discuss the importance of not replacing the instructor, but rather the use of technology to enhance the delivery style and learning opportunities of the participants [Jonassen et al, 1995; Reed et al. 1996; Ward & Lee, 1995].

Background

Computer Based Training (CBT)

Computer based training is a wide area which encompasses the development of software applications to engage the participant as an "active" learner. An instructor or software developer prepares "modules" of learning exercises for students to take at a later time and at their own learning pace. CBT modules may be used by firms to deliver a specific training lesson, whereby the student completes the module independently and then executes a test to demonstrate competency. An example would be new pilot "simulation" training or English grammar lessons. Shlechter [1990] uses CBT to train groups of military tank commander in enhanced messaging and communication skills. In the educational arena, a bright future is predicted for this "time flexible" service. An increase in enrollment of older aged students, and more students enrolling in full degree programs who have full time careers has placed demands on educational institutions for more flexibility of class scheduling in terms of time and location [Press, 1994].

More specific CBT examples are software learning tutorials. Stephenson [1992] uses a spreadsheet CBT to enhance the learning experience. Liegle and Madey

[1997] use CBT modules at a Fortune 100 company to train business professionals in HTML and web page development.

Distance Learning

Distance learning or distributed learning is a broader term, generally defined as the combination of two-way video and audio to multiple sites. In addition it may include computer aided instruction and additional multi-media tools to deliver course content [Burgess, 1994]. These additional multi-media tools encompass video clips, sound bites, visual transparencies, on-line tests and on-line exercises. Distance learning extends CBTs and permits the same training modules to be delivered at various locations. In addition, it encourages 'live' communications between the facilitator and the participants.

Not covered in this paper are other computer communication technologies such as e-mail, bulletin boards, and CD-ROM learning courses often included in distance learning options.

Delivery Changes and Effectiveness

Jonassen et al [1995] report potentially two-way interactive technologies are often used to present only one way lectures to students in remote locations. However the most valuable activity in a classroom is the opportunity for students to work and interact with each other and the facilitator. They believe that technology used in distance learning should facilitate "good learning experiences" in an extended classroom model rather then broadcast "teacher centered" lectures and demonstrations. In addition, their research shows that in a typical classroom the presenter or teacher contributes 80% of the verbal exchange, whereas in on-line computer conferencing the goal should be 10-15% of the message volume.

Ward and Lee [1995] report that distance learners have different needs than the students at the same site as the instructor. Distance learners know they will not be able to engage the instructor after class for private mentoring. They reported that distance learners ask more questions during the normal classroom setting. The implication is that the facilitator must provide time and encourage the distance learners to feel comfortable to request assistance during normal class on-line periods.

Initial studies have shown no decrease in learning outcomes and effectiveness at remote versus same site class participants [Patterson and Smith, 1994]. On the positive side, Kulik [1994] reports that students generally learn more in classes in which they receive some computerized based instruction. Kulik's work determined that students learn lessons in less time when utilizing computer based instruction. Another advantage of CBT is that students also like their classes more when they receive computer help and also develop more positive attitudes toward computers. However Kulik discovered some subject areas did not have positive effects from CBT. One area mentioned where there was not a positive effect from CBT modules was in general management courses. It appears that technical or skill training modules are very effective for CBT, while more theoretical coursework is not as effective.

Lajoie and Lesgold [1992] investigated the use of a computer training program to act as a coach for apprenticeship training. Their research shows that the computer can provide students with marked improvement in difficult troubleshooting skills with less overall learning times.

One of the overall benefits is the travel cost and time savings that distance learning sessions may provide. Participants can enroll in formal university courses and corporate training modules from their home or current work location [Ingram, 1996].

Success Factors

From research studies several factors have repeatedly surfaced that will increase the effectiveness of CBT and distance learning modules. Factors included are: student and instructor interaction, on screen feedback, effective screen designs, and a change in the role of the instructor. Alavi [1994] reports that no single or unified learning theory exists, but three attributes of effective learning processes can be readily identified. They include: active learning and construction of knowledge, cooperation and teamwork in learning, and learning via problem solving. She concludes that distance and CBT technologies can enhance all three of these learning keys.

Role of the instructor

Although CBT can deliver knowledge effectively, Stephenson [1992] reports that the instructor still has a role in computer based training. His research details that an increase in student and instructor interactions during CBT modules produces an increase in achievement. He compared the results of an experiment where an instructor walked around the room (mostly in an observation role) while students were completing a CBT software learning module, versus when the instructor sat in the front of the room and only answered questions when approached by a student. The first group had significantly higher achievement scores on post completion tests. Stephenson suggests that "there is simply something about having another human around and aware of your actions that alters your behavior." His implication is that a human instructor is necessary even where the course content might be significantly contained in the computer training modules.

Merely emulating the classroom learning experience should be the lowest possible level of acceptance in CBT and distance learning opportunities. Filipczak [1995] demonstrated that the use of communication and interactive technology permits engineers from industry to participate in the same classroom experiences as engineering students and university faculty. The need to develop artificial case studies is eliminated through the introduction of real business problems. The inclusion of professionals from the field as enrolled students in a traditional class enhances the learning experiences for all involved.

Jonassen et al. [1995] predicts the move from *teacher centered* to *learner center* approaches. Teachers are no longer the expert distributor of knowledge, and students are no longer passive recipients of information. Knowledge consists of shared understanding and intellectual breakthroughs with learning activities centering across authentic tasks that cross disciplinary boundaries [CAST, 1996].

<u>Skill level of participants</u>

Research has determined that the participants' prior knowledge of computer skills will influence the overall training achievement results [Stephenson, 1992]. The implications are that the participants must be comfortable with the equipment and "navigation technology" before learning a specific application. Stephenson suggests that subjects be trained on basic computer navigation and file management techniques prior to attempting a new specific training module.

<u>Feedback</u>

In a live classroom setting, the instructor can provide instant feedback to the participants, however in CBT applications, feedback is passive. It is often relegated to "your answer are correct" versus "your answer is incorrect". In a live classroom situation, a nod, a smile, or other confirmations from an instructor go a long way to motivate students in a classroom; feedback in CBT should have a similar effect [Cyboran, 1995]. Cyboran suggests that feedback is not necessary for every correct answer, but that enhanced feedback should include: letting participants know why their answer is wrong, using complete sentences to incorrect answers, and placing the feedback on the same location on each screen is vital. Enhanced feedback should not present new information in the feedback response; and complex or lengthy feedback statements should be avoided. In addition Cyboran states that the student should be given the correct answer after a reasonable number of incorrect attempts (suggests three maximum).

<u>Computer based testing</u>

Bugbee [1996] reported that the computer can be used for effective test giving. Computer tests have been shown to reduce testing time, to obtain more information about the test takers, to increase security, to provide instant scoring, and to be more easily scheduled than paper and pencil tests. However Perkins [1995] details a study that found that hypermedia testing can cause different levels of anxiety and performance than paper testing. Prior knowledge and working experience of the computer as a tool is vital to reduce these factors. This implies that courses in which the students have limited or no prior experience with computers should *not* be given computer based testing.

<u>Overall Development Techniques</u>

Considerable time and effort should go into the planning and development of a training module on a computer. Especially important is the concept of a non-linear approach to a lesson. This non-linear approach permits the student to control the flow of the lesson to some degree [Molina, 1995]. Molina details the steps of an effective CBT to be:

- gather the information from the field,
- identify what has to be taught based on learning deficiencies
- use objectives in design document to begin writing rough instructional strategies
- prototype to see how the script will look and sound to the user.

Other considerations are detailed by Jonassen et al. [1995], where they detail that the computer provides the opportunity to develop problems for students that are either replicas or analogous to the kinds of real world problems faced by citizens, consumers, or business professionals. These problems should require from the student a repertoire of knowledge, and the appropriate application of knowledge and skills to prioritize problem classification and solution phases. They provide challenging situations for students to work together to solve real problems as contrasted to the traditional classroom where the amount of resource information is not available. The impact of using a variety of media style cannot be underestimated. Ward and Lee [1995] report the average adult attention space is twenty minutes, therefore a combination of slides, overheads, audio, video, hands on exercises are vital in design considerations.

In the design phase, the overall flow and the degree of independence given to the student is important. In an application that is truly independent, the learner is given the navigation tools to skip or choose the modules. However on the negative side of independent navigation, particular learning goals might be avoided by students.

Garrison [1989] reports a mixture of independence as well as control is desired. Learners can choose their learning path, once evidence of competency has been shown in particular areas.

Two way video considerations

In distance learning opportunities, Fish et al. [1993] report that spontaneous informal communications is useful. It is a crucial method that organizations rely on to accomplish work, transmit organizational culture and knowledge. They believe that the use of live video in learning opportunities will increase the spontaneity and frequency of communications, and support social relationships. Two-way video helps to cope with the most complex and equivocal communications problems encountered in work groups and as a result, integrating member into and supporting the work in research and development groups. In typical distributed learning applications, two-way audio and videos allow for increased dialog from outside experts into the classroom.

Group interactions

Shlechter [1990] discovered that CBT training when mixed with group learning techniques enhanced subject learning and retention rates. He assigned sixty four students in three different experiments where students worked independently on a CBT application versus groups of three or four students on one PC terminal. Overall the group training subjects consistently completed the courseware more quickly and with less difficulty than did the individual training subjects. Also the group training subjects learning was shown to be more resistant to forgetting. His conclusion is that CBT group training was more cost effective than individual CBT training by a factor of five.

Alavi [1994] also reported a study in which a Group Decision Support Systems (GDSS) was used to enhance traditional classroom learning. She concluded that GDSS collaborated learning leads to higher levels of perceived skill development, and also higher final grades by the students in the computer supported study.

Measurement Techniques

As with any investment of time and resources, it is important to have effective means to measure the overall learning impact of the CBT and distance learning technologies. A review of the research shows various means are used to gauge the effectiveness ranging from pre and post paper tests, on-line tests, expert observers, and utilization of learning logs.

Overall Beattie [1994] offers guidelines for the evaluation of CBT software used for learning experiences. Her research involves developing clear research questions. They are:

- identify the task [what is the problem?]
- does it work?
- are you sure someone needs this software
- what do you want to teach
- determining the appropriate research questions to determine what really going on.

Clark [1994] details three overall points in the design of measurement techniques. First evaluation considerations should be incorporated at the start of the distance learning problem. This is to assist in building evaluation techniques into the overall module design. Second, all programs should adopt a multilevel evaluation plan, and include a combination of questionnaires, diaries, open ended participant reactions, and quantitative student achievement scores. Third, delivery cost should be monitored and evaluated.

Paper pre-test and post-tests

Many of the experiments which involved CBT or distance learning, used paper and pencil pre and post exams to measure knowledge learned. A paper system was used to avoid the problem of various degrees of computer usage skills by the participants effecting the actual learning results.

Shlechter in 1990 used paper and pen tests prior to the initial CBT training as well as immediately following the training; another test two weeks following completion; and a third one month following completion of the training. His overall results demonstrating increased retention and group skills utilizing group CBT are detailed previously.

Participant questionnaires

At Southhampton University [Dobson et al. 1995], a "hands on" electrical engineering laboratory experiment of building amplifier electronic circuits for an application was replaced by a computer simulation laboratory assignment. Students were segmented into two groups, one using the computer simulation, and one employing the "hands on" laboratory technique. Questionnaires were developed for participants to measure their computer literacy, attitudes toward computers, and ease of completion of the assignment was measured. Interestingly they found that the computer enhanced the learning effectiveness *only* to a certain point, however that the "hands on" laboratory experiment was still necessary. Their findings conclude that the a mixture of "hands on" and computer simulation was the best technique.

A post course questionnaire was developed by Alavi [1994] where she measured the effectiveness of a group learning process. She used a five point Likert scale to measure all items. It attempted to separate the responses for learning and knowledge gained versus evaluation of the actual computer software package utilized. Those completing the computer assisted sessions reported higher levels of skill development, learning, and interest in learning relative to students not enrolled in the computer assisted training sections. Finally she compared the "final grades" in the course for the students utilizing computer assisted students versus students in the non assisted classroom. Those students with computer tools consistently had higher grades.

Expert observers and learning logs

At the University of Glasgow, Draper et al. [1996] completed a two year study where expert observers and evaluators were employed to measure the effectiveness of newly developed CBT modules. This expert team developed instruments, pre and post tests, detailed interview questions with instructors and participants, and direct observation of class session evaluations. These experts were not involved in the development or delivery of the training modules. They were kept separate from those decisions, and were brought in only to evaluate the effectiveness of the modules.

Evans [1994] recommends interactive logging as a research method. He claims the computer can automatically gather data for the researcher without intervention and interruption to the participant. However vast amounts of data can be logged without knowledge of what task the participant is actually doing. To overcome this limitation he recommends additional observation, field notes, and video.

Fritze [1994] also endorses the use of interactive logging of user data as an evaluation technique. He identifies the people involved and the data to be gathered into the following categories:

- the planner to justify and allocate resources
- the content designer to design and change curriculum and effected format of the content
- the facilitator to identify problems associated with usage of the material
- the student advisor to identify students at risk or to assist students in adopting appropriate study strategies
- the instructional framework designer to identify navigational and interactivity patterns and problems
- the educational researcher to confirm and develop models of student learning
- the learner to receive feedback on performance and advice on appropriate strategies.

Difficulty in developing evaluation techniques

Bernthal [1995] offers some suggestions to overcome developer bias when creating measurement techniques. They include: identify the organization's values and practices, identify skills, knowledge and attitudes, define the scope and purpose of the evaluation, identify data sources, choose the best method for collection data, and select the measurement approach. Questions to be asked include: how frequently do you want to collect data; would data from a control group strengthen the findings; how many people should I collect data from.

Summary

In conclusion, CBT and distance technologies expand the learning options for both organizations and educational institutions. CBT can provide trainees the opportunity to learn at their own pace while offering flexible learning times. Distance learning options with two way audio and video provide increased dialog opportunities between participants and facilitators. Travel cost and time savings are benefits spelled out in many articles.

It is clear that learning objectives must be spelled out in advance and that the traditional lecturing style of teaching is not effective in distance learning opportunities. It has also been demonstrated that group learning situations on a personal computer can speed up the learning process and increase retention. In the design stage, consistency of design and screens was stressed.

The overall success factors most frequently mentioned are in the design area: 1) clear definition of goals, 2) consistency of screen layout is very important, 3) leaving of white or clear space is necessary, 4) proper and enhanced feedback increases learning, and 5) utilize more then just text, add video, audio, and interactive testing in module design

In the delivery arena: 1) a move from teacher centered to learner centered, 2) built time into the modules for increased questions, especially from remote site participants, 3) the availability or presence of an instructor during CBT modules enhances learning, 4) group learning is more effective than individual learning modules, and 5) insure that participants have basic computer skills prior to delivery of a specific training module

Measurement techniques include: 1) pre and post paper tests are the most popular method, 2) computer based evaluation techniques may be employed if the participants have previous computer experience, and 3) expert observers, combined with automatically logging of responses by participants may be an effective measurement technique.

References

Alavi, M. Computer mediated collaborative learning an empirical evaluation, *MIS Quarterly*, **18 2**, June 1994, 159-174.

Beattie K. How to avoid inadequate evaluation of software for learning, *Interactive Multimedia in University Education*, IFIP, 1994, 245-257.

Bernthal, P. Evaluation that goes the distance, *Training & Development Journal*, **49 9**, Sept. 1995, 41-45.

Bugbee, A. Equivalence of paper and pencil and computer based testing, *Journal of Research on Computing in Education*, **28 3**, Spring 1996, 282-299.

Burgess, W. *The ORYX Guide to Distance Learning*, New York: Oryx Press, 1994.

CAST, Center for Advanced Study in Telecommunications, Ohio State University, Columbus, OH: Winter 1996, **1**, 1-6.

Clark, R. Assessment of Distance Learning Technology *Technology Assessment in Education and Training*, Hillsdale, NJ: L Erlbaum Associates, 1994, 63-78.

Cyboran, V. Designing Feedback for Computer-based Training, *Performance & Instruction*, **35 5**, May, 1995, 18-23.

Dobson, E. L. Hill, M. & Turner, J. D. Evaluation of the student response to electronics teaching using a CAL package, *Computers Education*, **25 1**, 1995, 13-20.

Draper, S. Brown, M. Henderson, F. & McAtee, E. Integrative evaluation and emerging role for classroom studies of CAL, *Computers Education*, **26 1** 1996, 17-32.

Evans, P. Investigating the use of interactive hypermedia systems, *Interactive Mulitimedia in University Education*, IFIP, 1994, 259-265.

Filipczak, R. Putting the Learning into Distance Education *Training*, **32 10**, October 1995, 111-117.

Fish, R. Kraut, R. Root, R. & Rice, R. Video as a Technology for Information Communication *Communications of the ACM*, **36 1**, Jan. 1993, 48-62.

Fritze P. Visual Mapping approach to the evaluation of multimedia learning materials, *Interactive Multimedia in University Education*, IFIP, 1994, 273-283.

Garrison, D.R.; *Understanding Distance Education*, London, NY: Routledge, 1989.

Ingram, A. Teaching with Technology, *Association Management*, **48 6**, June 1996, 31-36.

Ives, B. & Jarvenpaa, S. Will the internet revolutionize business education and research, *Sloan Management Review*, Spring 1996, 33-40.

Jonassen, D. Davidson, M. Collins, M. Campbell J. & Haag, B.B. Constructivism and Computer Mediated Communication in Distance Education, *American Journal of Distance Education*, **9 2**, 1995, 7-25.

Kulik, J. Meta Analytic Studies of Findings on Computer Based Instruction, *Technology Assessment in Education and Training*, Hillsdale NJ: Lawrence Erlbaum Associates, 1994, 9-33.

Lajoie, S. & Lesgold, A. Apprenticeship training in the workplace: computer coached practice environment as a new form of apprenticeship, *Intelligent Instruction by Computer*, Wash. DC: Taylor & Francis, 1992, 15-32.

Liegle, J. & Madey, G.; Investigations into Technologies for Distributed Learning, *Proceedings of the Midwest DSI Conference*, April 1997.

Molina, C. Transitioning to CBT", *Performance & Instruction*, **35 9**, October, 1995, 26-33.

Patterson, J.S. & Smith, W. North Carolina Information Highway, *IEEE Network*, **8 6**,. Nov/Dec 1994, 12-17.

Perkins R. Using hypermedia programs to administer test: effects on anxiety and performance *Journal of Research on Computing in Education*, **28 2**, 1995-96, 209-220.

Press, L. Tomorrow's Campus, *Communications of the ACM*, **37** 7, July 1994, 13-17.

Reed, W. Spuck, D. & Bozeman, W. Accessing the impact of computer based learning since 1987, *Journal of Research on Computing in Education*, **38 1**, 554-557.

Stephenson, S. The Role of the Instructor in Computer-Based Training, *Performance & Instruction*, **32 7**, August 1992, 23-26.

Shlechter, T. Relative Instructional Efficiency of Small Group CBT, *Journal Educational Computing Research*, **6 3**, 1990, 329-341.

Ward, E. & Lee, J. E. An instructors guide to distance learning, Training & Developmen, 49 11, Nov. 19, 40-44.

Establishing a Hands-on Lab Approach for a Course in Local Area Networking with Emphasis on Problem Solving

by
Mark W. Smith, Ed.D.
Associate Professor, Computer Information Systems and Technology
Department of Computer Technology, Purdue University

A major difficulty with teaching a course in Local Area Networking (LAN) is the development of a meaningful lab component. Variables such as the instructor's depth of knowledge with the various LAN hardware, Network Operation Systems (NOSs), and synchronization of lecture to lab, combine with the students' varying levels of skills and knowledge, hands-on experience, and numbers of students per lab, requiring an innovative and flexible paradigm for LAN lab construction. The focus of this tutorial is to help the educator design and implement the lab portion of the LAN course so that the student comes away from the lab with two basic skills: hands-on experience and improved problem solving skills. The hands-on experience consists of the student actually installing and configuring the hardware components, such as Network Interface Cards (NICs), cabling, and networking devices, along with the software installation of the server-side NOS, client-side software, and necessary drivers, in order to configure and implement a working LAN. The problem solving component is actually a process which is first introduced in lecture and then refined during the semester's various lab assignments called experiences. How to coordinate as many as 75 students, over 20 two- or four-week lab experiences, with limited hardware/software assets will be demonstrated along with the documentation, such as assignment sheets, experience guidelines, and evaluation criteria necessary to implement this unique strategy.

As mentioned, the lecture portion of the course is de-coupled from the lab. While the lecture deals with concepts such as network architectures and protocols, the lab portion's major goals are development of the student's problem solving, coping, and team skills along with an emphasis on project management and reporting techniques, all experienced in a hands-on environment. However, to emphasize the importance of developing one's problem solving skills, a successful two-lecture format describing what problem solving is, and how it is applied, has been taught during the first week of lecture and is included in the tutorial materials.

Initially designed by the author's collegue, Professor James E. Goldman, the lab portion of the course has continued to evolve into a series of lab experiences based upon available hardware and software. Through purchase, but mostly through gifts/grants of various hardware and software vendors, approximately 15 to 20 different experiences were developed. Labs consist of approximately 15 students which allow for three teams of five members for each lab, and four or five labs per course. The key to success has been in the organization and planning stages of the experiences prior to the beginning of the semester. Student teams are assigned five two-week experiences and one four-week experience providing a range of experiences for each group. All teams must complete the wire and cable testing experience, with each team also assigned two general connectivity, two peer-to-peer, and one client/server NOS experience for the semester (allowing an extra week at the beginning of the semester for initial setup, and a final lab for tours of real-world LANs). The Where to Start Sheets (experience guidelines), and the assignment grid will be discussed in the tutorial and are part of the tutorial materials package.

During the two hour lab periods, the teams must organize, plan and conduct experiments with the particular technology of the assigned experience. Teams are expected to put in extra hours outside of their normal lab in order to push the experience to the limit. The instructor acts as a facilitator and manager. When a team has a problem, the instructor helps them to develop their problem solving skills by asking the right questions, such as what are the knowns and unknowns in the particular situation. When the two weeks are up, a report must be written for the experience. The report details the team's accomplishments, but in particular, must detail the problem solving steps taken when problems were encountered. Report guidelines and grading criteria will be discussed and are included in the tutorial materials with sample reports available at the tutorial session.

Overall, this methodology, while at first frustrating for the student, ultimately helps the student to develop a strong hands-on problem solving process which can be taken out of the course and immediately applied to his/her new job. This approach to conducting LAN labs emphasizes education by teaching how to cope with networking issues and apply problem solving techniques, rather than a training approach which merely instructs the student on what buttons to push and how to use a particular technology which will change within six months.

BUILDING GLOBAL IS QUALITY EDUCATION THROUGH FACULTY-BASED CONCEPTS

Herman P. Hoplin
School of Management
Syracuse University
Syracuse, NY 13244-2130
(315) 682-2393; Email: hphoplin@mailbox.syr.edu

ABSTRACT

Information Systems (IS) and Information Technology (IT) have gone global. Globalization is causing shifts in IS concepts and more focused education. We now think in terms of global economy, competition, and perspectives--and we should also think of IS education as global. Max Hopper points out that going global requires a long-term horizon and executives need to be aware of the manner in which their business can go global. (Hopper, 1995) We must view IS education in the same context. This presentation will serve as a guide to do this and to encourage IS faculty to exercise leadership to achieve innovate IS concepts leading to quality education.

INTRODUCTION

Implied in the title of this paper is the thought that there is a need to build quality global Information Systems education on the precepts and beliefs of professional IS faculty. Faculty, active in the field, are facing the realities that confused and inferior education at almost all levels is being doled out to students and audiences who are desperate to moved ahead and keep pace with the information explosion and increasing demands of business and society. The situation requires keeping up with advances in technology and putting skilled leadership to work to solve today's and tomorrow's problems.

Existing conditions dictate the need for revisiting IS strategy. Such revisiting readily reveals that IS is no longer predominately a technical problem but rather a management problem requiring the use of technology and computers to solve the problem on a timely basis. Peter Drucker points out that knowledge workers of today and tomorrow must be willing and even eager to buy into the company's mission and that both the hard and soft disciplines are needed. He makes an analogy that just as one needs two hands (one for running a business and the other for building an organization) where each hand must know what the other one is doing, and that both must say the same things; otherwise management and the company will lose all credibility. (Management Review, Oct 1995) This is indicative of the need for management by corporate objectives which are now requiring IS to broaden its goals to the enterprise level.

Drucker is not the only management authority to see the need for revisiting IS strategy. In the education or curriculum wing of the thought process, Warren Bennis pointed out as far back as 1989 that our educational system is slow in recognizing the potential power of IS. Bennis stated that our educational system is better at training than educating and that we need people to find problems are not always clearly defined and they are not linear. (Bennis, 1989) Within the last four years, Gordon Davis, in receiving DPMA's Education Foundation Educator-of-the-Year Award, commented that introducing vendors' software into the classroom stops the conceptual education process. (ISECON, 1992) There is a veiled threat here that we are not educating our IS students--only training them.

Rather than "fighting" technology advances, IS faculty and managers in the field should view these advances as giving them new ways of providing service to users, customers, or other beneficiaries of IS. This can also result in new strategic benefits and become a redefinition of IS for customer service. IS faculty should always empower their students by providing higher-quality/value information to lead to better applications, improved response times, and reduced cost to their employers when the students have turned their efforts to job related endeavors. IS faculty, to survive over the long pull, must provide their students with information that the students future executives need for business survival. Drucker points out that it is the concepts, not the tools that are important. It is those concepts that are needed to understand how to use information in businesses that convert into techniques and into integrated IS. (Drucker, 1995)

Without much navigating, frequent changes in today's technology are strongly indicating that tomorrow's technology is emerging as something entirely different from what we have seen to date in the era of computing and information systems. This paper will report on several concepts and process improvements that are rapidly overriding legacy information systems that are entrenched in the private, public, and educational environments.

GLOBAL IMPACTS

The information explosion and increasing business and environmental demands on data processing have not only gone global but, in so doing, have put heavy pressures on IS education and management. For competitive global advantage, successful businesses have been able to partially integrate the changing IS environment into the business environment. However, to accomplish this to a greater degree requires a bonding of business and IS goals. Without widespread IS faculty leadership IS specialists are finding this like steering a ship without a rudder. To bring about the philosophical change needed to accomplish this requires that managers at all levels have an IS educational background that is so far strangely absent. Successful managers in search of excellence have found out that they can no longer expect to find magical solutions to technical problems. This results in an awkward situation that can be corrected only in faculty peer-based solutions if we are to successfully meet the next millennium. We as IS educators must face this as an urgent challenge.

This challenge is sharpened by statements such as one made by Peter Drucker in 1991 that, "within the next 10 years, both the structure of organizations and jobs of senior people within organizations will be drastically changed, primarily because of information." There is little doubt that change, and with it flexibility, is the key to success in IS management. The rapid proliferation of emerging information technologies drives home the point that IS cannot run in place without losing ground. Even the traditional classes of architecture for data processing, word processing, and data management, spread sheets, and data communications are changing at a greater pace than current IS organizations can absorb.

To meet today's and tomorrow's challenge requires that IS faculty and external management look at the entire enterprise--not just the technical. This involves the global business environment and processes, and the deftly fitting in of quality and reengineered information technology (IT) to meet increasing demands in the prevailing environment of greatly reduced budgets. Taking this broader view does not lessen the impact of automation and high technology but shifts some of the emphasis in other directions. The focus can remain on the computer but, in the final analysis, values to the organization are in the business processes. Only with the shift to a business philosophy can bridging the gap between research and practice become a reality in the 21st Century. To further analyze this in this paper, ten major emerging technologies that must be recognized by IS will be identified as critical and trendy developments.

To review briefly the relatively short history of computer-based IS and put it in context with global education, professionals have gone through a series of generations which were initially based on heavy technical development of hardware, soon followed by software, and later by peopleware. Early developments were focused on programming, statistics, and data processing. Later, preoccupation with getting the computer to work gave way to concerns about what functions computer systems should do including numerous applications in which high technology would serve business as well as research and development operations.

IS soon became a feature of automated systems where word processing, decision-support and decentralized processing developed into major courses of action in carrying out business. The digital computer opened the door to replacing electrical accounting machines (EAM) and in a generation or two this lead to office automation, microcomputers, and multiple niche applications as we know computing today. Greater demand for applications, some of them global, led to the need for integrated systems such as computer-based information systems, networks, and webs based on the idea of serving management and the environment at all levels.

As some authors have found, it is useful to consider IS as a business within a business even though integrating IS into the other functional areas of the organization may cause special organizational and strategy-formulating challenges. (McFarlan and McKenney, 1983) However, this integrated approach must be taken in order to address the organizational issues surrounding IS. This paper focuses on this approach which is drastically changing the nature of the IS field. Notable among these trends is the use of IS technology as a part of corporate strategy, end user computing, and the use of PC's as managerial workstations. (Lucas, 1990) Networking and its extensions such as email and the World-Wide Web in the last few years have rapidly propelled IS onto the world-wide scene which is continuing to expand exponentially with each passing day.

EDUCATION AND RESEARCH FOCUS AREAS

Seven education/research areas as indicated by ▪ are briefly discussed below:

▪ **Visionary Leadership for Continuous Improvement**--If you work in a university setting, you are in the learning business. Facilitating student learning effectively, is vital whether it be IS or in other fields. It requires visionary leadership and a goal of quality education. IS faculty need to inject this leadership into their classes for the field to succeed.

▪ **Revolutionary Thinking Through Peer-based Development**--Although often shied away from, to reach needed IS initiatives, revolutionary thinking is required and should lead to integrating ingenuity and quality through peer-based faculty development. Initiatives such as reengineering is sometimes needed to make a course or department come alive.

▪ **Major Emerging Technologies**--Ten major developments are listed below as being indicative of emerging challenges to our global environment:

❶ Distributed Processing Including Data Communications and Networking.

❷ Windows and Client/Servers.

❸ Modern DBMS including Object-Oriented Systems.

❹ Impacts of Stakeholders and End-User Computing.

❺ Bringing Languages into the 21st Century.

❻ Use of Automated Tools.

❼ The Data Highway (World-Wide Web).

❽ Expert Systems, Neural Systems, and Virtual Reality.

❾ Electronic Data Interchange (EDI).

❿ Other Types of Connectivity Yet to be Developed.

▪ **TQM and Reengineering**--These two areas have many similarities but also have significant differences. Quality has many dimensions. The aspect most commonly associated with quality is the absence of defects (Previously referred to as Zero Defects in programs fostered by the Department of Defence). Increased quality is enhanced with software reusability which require more testing and quality assurance than nonreusable components. This tends to align performance management with corporate goals. (Yourdon, 1994) Quality (TQM) programs work within the framework of an organization's existing processes. Experts claim continuous incremental improvement through the aim of doing what we already do, only doing it better. (Hammer and Champy, 1994) Reengineering, on the other hand, advocates a radical change in management and management processes, and seeks breakthroughs, not by enhancing existing processes but by replacing them with entirely new processes. (Hammer and Champy, 1994) Both of these initiatives have their place and can assist in better IS education.

▪ **Ubiquitous Nature of IS/IT**--This is perhaps the broadest of focus areas since it epitomizes how quickly IS/IT has spread into every nook and cranny of organizations, and for that matter, the global environment. IS no longer belongs to an organizational niche; no one can claim it; it is universal. But it still needs to be managed by knowledgeable IS people who have broad concepts of making IS available to all managers and users . This is probably the greatest challenge to IS educators. There is no room for building technical empires; all efforts have to be in the interest of making IS available to all legitimate managers, clients, and innovators. IS impacts on individuals and organizations on a global basis. With focus on information systems (IS) and information technology (IT) the joint functions assert the need for a stronger orientation, a process-oriented approach requiring great ingenuity and adaptability on the part of IS educators and practitioners. Since the advent of the "net" and "web," it has become increasingly clear that the focus of IS may have been inherently wrong. IS people are often obsessed with structure when they should be more applications-oriented. The focus should rightfully remain on the computer but not at the expense of placing technology in the driver's seat when the real values of the technologies are in the business processes. Computer professionals are now faced, not only with developing connectivity between mainframe and PC systems, but they are now having to deal with the burgeoning area of communications which is sometimes referred to as networking. In an article in the *Harvard Business Review* on "How Networks Reshape Organizations--For Results," the point is made that "a network identifies the 'small company inside the large company' and empowers it to lead.

(Charan, 1991) This is a powerful challenge to IS educators to provide the educational leadership to accomplish this.

- **Communication and Automated Tools--** Universities and Colleges are under pressure to teach state-of-the-art tools in IS programs and to prepare graduates to learn new tools of the trade. Client/Server, Object-Oriented methods, and distributed processing including networking are three new technologies that need to be included in the IS curriculum. Maintaining up-to-date IS curricula are serious problems since the turnover in the field is now two years (or less)--not five years as it was a year or two ago. This rate of change management is likely to continue. Almost all business functions are going electronic. On the down side (for pure academicians) the education process is almost becoming "education on demand." There is some argument over the use of automated tools in high technology. For example, an article on "Data Mining and Visualization"(May 1996) pointed out that the goal of data mining is to find actionable patterns in such data using automated tools. This would be a reasonable and effective way of attaining this goal. A reader from industry took exception to the article as an attack on statisticians and traditional analysis methods. The author pointed out that many IT professionals have and continue to fight the battle against those who would replace programmers with automated program generation. Although there are two sides to the issue, it appears that the "Luddite" view of the reader is becoming overshadowed by the use of automated or semi-automated tools where the computer can be of greater help in problem-solving. (Weldon, 1997) The IS challenge is to help people work together more effectively in easing frustration and conflict caused by rapid technology advances. Vendors including Learning Companies are now developing plans and profiles to accomplish this. (Carlson, 1997)

- **Dimensions of Leadership--**IS faculty need to expand in the area of leadership if IS is to remain as an effective field. Hard work at the computer with our heads "under a bushel" does not, by itself, sell and proliferate the field. We must discover how we want to lead and who the targets are--first our students but our eye must be on the employers of our students. Consulting and educational firms are now developing approaches to understanding and developing effective leaders. We must take advantage of this process. One learning company has created an effective analogy on leadership and nature pointing out that when geese fly in formation, the leader does not remain the same. Several of the flock take turns moving to the front of the vee. As the leader tires, he(or she) falls back in the formation as a follower while the leader encounters the air currents and continues the course. (Carlson, 1997) This seems like good advice for us in the IS faculty. This strategy would not only avoid burn-out but spreads the leadership experience to others in our peer group to take advantage of IS contributions that would not otherwise surface,

CHANGE IN ORGANIZATIONS AND THE ROLE OF IS

As many in attendance at this conference will attest, managing change in IS organizations appears to be the key to success in both large and small organizations. Researchers have found that even with an organization's survival at stake, getting an organization to change is a difficult task. Change primarily occurs because of external (environmental) pressure rather than an internal need to change. (Goldstein and Burke, 1991) Unless organizational change (either academic or industry) can be done successfully, the future of the organization will be short-lived.

In addition to the "change" challenge, education and management authorities see our organizations heading in different directions. Drucker sees the organizations of tomorrow flattening out. There will be little if any hierarchy and the employees will become specialists, much like we see in a hospital. He sees information becoming more important and even calls it the information-based organization. There will be multiple factors causing this change, but Drucker believes that information technology will drive the shift. He discusses the shift from data processing to information processing and the resulting organizational structure changes. (Drucker, 1988) This viewpoint seems to highlight the importance of identifying critical success factors. Ralph Carlyle, on the other hand, believes that centralization will return and that, while there will be specialists, the "organization will be eager to spawn multidisciplinary generalists. (Carlyle, 1990) Carlyle recognizes the impact that Information Technology can have on the organization in that he believes that Information Technology will be an enabler of change and that the change will be the result of what business professionals want.

Although these views differ, they both may be correct. The implications of these differing views are multifaceted. First, IS faculty and managers of the future will be faced with newer and even more difficult challenges. These will include selecting the appropriate technology for an organization and to become IS well-rounded faculty and managers rather than basically technicians. Second, the future role of IS is ambiguous at best. It may enable the needed change or cause it. Whatever the case, as organizations and the people in them evolve so will the role of technology in the organization.

From an educational point of view, IS curriculum changes may be rampant. It will have increased breath in some topics, more depth in others, and yes, faculty retraining will become a requirement. Many university constraints and potential conflicts with AACSB will complicate updates to meet the needs of graduates in the real world. Networking will become the basic part of IS programs along with increased management issues and security and privacy concerns. Multimedia and graphical application development will expand with increased industry requirements and enable more user friendly applications. The increased scope of IS curriculum collaboration with other fields and departments will become the order of the day. Joint committees of ACM, AITP, ICIS, and AIS are already developing this collaborative concept.

IMPLICATIONS AND CONCLUSIONS

So what does all this mean for the IS faculty member, the IS professional, and the IS organization?

❶ The IS person is in a multi role--in a professional transition moving from technician to manager with considerable exposure as a trainer and educator. There is a responsibility to educate the organization about the new technology available and how it can support the organization's goals. IS professionals must educate management and users to increase their awareness of the strategic importance of IS.

❷ The fundamental problem is one of designing strategies to assist management in the use of information technology to best accomplish and expedite the organization's goals and objectives.

❸ New technology provides alternative options to accommodate the complexities of various goals, different organizations' cultures, and myriad personal styles. Each organization must identify its own goals and the process through which they can be achieved. IS systems professionals must be able to do or enable this.

❹ Recognizing and accommodating human responses to change and conflict is paramount. If common organizational goals are established, an environment of cooperation is more likely to exist. No matter what approach is taken, there will be resistance to change and some conflict. The role of the IS faculty or professional becomes one of a facilitator or catalyst as the organization evolves through the process.

❺ The organization's goals (academic or business needs) must be considered as the basis for change and to use technology options to accomplish this.

❻ As technology becomes more widespread in the organization, IS must involve users more and more.

❼ Because IS is in a rapidly changing state, much additional research is needed in this decade if we are to bring about success from introducing innovative computer-supported technologies.

❽ In view of rapid developments, both on and off the web, hopefully we have now received the "wake-up" call and that we will take great strides to educate and improve IS people and organizations to meet the IS challenge of the late 90's and into the next millennium.

❾ Most assuredly we have now reached the point, both regionally and globally, where IS is not only ubiquitous but is universally needed--IS is now a global problem crying for strong IS faculty-based leadership.

References

Bennis, Warren. *On Becoming a Leader*. Reading, MA: Addison-Wesley Publishing Company, 1989.

Carlson Learning Company Brochure. *Personal Profile System* "How to Help People Work Together More Effectively, 1997.

Carlson Learning Company Brochure. *Dimensions of Leadership Profile*, 1997.

Carlyle, Ralph. "The Tomorrow Organization," *Datamation* (Feb 1, 1990), pp. 22-29.

Charan, Ram. "How Networks Reshape Organizations--For Results," *Harvard Business Review* (Sep/Oct, 1991), pp. 104-115.

Davis, Gordon. *ISECON '92 Program*, Nashville, TN (Oct 17-18 1992).

Drucker, Peter F. "The Coming of the New Organization," *Harvard Business Review*, (Jan/Feb 1988), pp. 45-53.

Drucker, Peter F. "The Information Executives Truly Need," *Harvard Business Review* (Jan/Feb 1995), pp. 54-62.

Drucker, Peter F. "Drucker Speaks His Mind," Interview by Mike Johnson. *Management Review* (Oct 1995), pp. 10-14.

Goldstein, Leonard D. and W. Warner Burke. "Creating Successful Organization Change," *Organizational Dynamics* (Spring 1991), pp. 5-17.

Hammer, Michael and James Champy. *Reengineering the Corporation.* New York: Harper Business, a Division of Harper Collins Publishers, 1994, p. 49.

Hopper, Max D. "Its Impact on Globalization," *SIM Network* (Dec 1995), pp. 1,2,5.

Lucas, Henry C. *Information Systems Concepts for Management.* 4th Edition. New York: McGraw-Hill, Inc., 1990, p. 12.

McFarlan, F. Warren and James L. McKenney. *Corporate Information Systems Management: The Issues Facing Senior Executives.* Homewood, IL: Richard D. Irwin, Inc., 1983.

Weldon, Jay-Louise. "Automated Tools Can't Replace a Good Statistician," *Database Programming and Design*, (Feb 1997), p. 9.

Yourdon, Edward. *Object-Oriented Systems Design: An Integrated Approach.* Englewood Cliffs, NJ: Prentice-Hall, Inc., Yourdon Press, 1994, pp. 22, 67, 91.

Information Systems Education: Introducing Cultural Aspects of Systems Development

Rick Gibson
American University
Department of Computer Science and Information Systems
4400 Massachusetts Avenue
Washington, DC 20016
(202) 885-2735
rgibson@american.edu

Abstract

Although most universities have already undertaken efforts to include international components in their general educational requirements, most information systems departments have chosen to ignore the important issues that surround the use of information technology in a global context. This paper presents some issues that should be taken into consideration so that globalization and cultural factors can be seamlessly integrated with information systems development course work.

Introduction

Among the forces exerting pressure on organizations, globalization is perhaps the single most significant in terms of both opportunities and challenges. The opportunities include new forms of worldwide commerce and markets, along with new forms of inter-organizational relationships to enhance productivity. The technical challenges center on the development of information systems that dramatically increase connectivity, which is enabled by extensive communication networks.

The information systems curricula at most universities were developed before globalization fundamentally altered the knowledge and skills required for systems development. Granger et al. (1992) contend that increases in technological innovation and globalization of the marketplace are two forces that need to be emphasized when redesigning the information systems curriculum. While not suggesting that internalization must be a component of every course, they consider the topic to be one that graduating students have mastered rather than merely surveyed.

Deans (1992) conducted a workshop on international information systems curriculum development with three goals: awareness (e.g., infusion via a lecture on transborder data flows), understanding (requires a formally trained instructor), and literacy (with foreign language and political science courses). She advocates a natural progression from theoretical to practical to curriculum levels of change. In a panel discussion chaired by Palvia (1992), the several graduate programs (e.g., Arizona, Texas, Memphis State, Georgetown, Penn State) already offering a single global IT course were compared. A comparison of these graduate programs revealed a lack of consensus, with some schools offering either a single global information systems course or global topic coverage in one or more or all core classes. The critical learning factors that emerged were awareness, communication, collaboration, ethics and continuous learning.

In spite of recent efforts such as these, there appears to be a serious shortage of information systems professionals who are proficient at understanding and operating in the global marketplace. International software must be as functional in its foreign market as it is in its domestic market. The failure by both business and information systems researchers to offer managers viable options for designing, selecting and transferring the software resources needed to create international information systems inhibits firms from moving aggressively in global markets.

Fernandes (1995) contends that the world dominance of American software is enforcing a new form of imperialism, and that software should be *localized* for ethical and market reasons. Uren et al. (1993) support the economic impetus for localization by suggesting that although early adopters of software products tolerated software in English, with Western cultural bias, mass markets demanded that products are seen in their own language and firmly based on their own culture. Frye (1997) describes the addition of global capabilities to the functionality offered by software vendors as a paradigm shift on par with the late 1980s shift to client-server architectures. The challenges associated with development of international software are twofold, emerging from concerns about effectiveness and efficiency. The primary challenge is to deliver additional languages and localizations that effectively satisfy local requirements.

Localization

A *locale* describes the user's environment—the local conventions, culture, and language. Conceptually, it is possible to consider a locale as made up of a unique combination of language and country, e.g., English/United States or French/Belgium. *Localization* is the process by which a software application is adapted to its locale.

Uren et al. (1993) describe localization as an aspect of software engineering that emerged from an awareness of the complexity of re-writing. Localization is a broad term that takes into account all aspects of the international problem both design and technical. The term refers to the process of making changes to a globalized product to make it useful and viable in a particular market. Since the term is often misunderstood, this adopts a refinement suggested by Fernandes (1995), which uses three categories of localization: technical, national, and cultural.

Technical Localization

The most straightforward localization tasks reside within the domain of software engineering and include concerns about: double-byte character sets, arithmetic operations, sorting, string comparisons, memory, modems, and operating systems. Many operating systems and compilers for fourth generation programming languages include features that provide extensive support for solving these technical problems associated with developing international applications.

National Localization

Accommodating national differences in punctuation, page size, dates and monetary symbols allows users to concentrate without the distraction caused by the need to accommodate styles other than their own. More importantly, this category of localization includes the formal process that makes a program written for one language freely usable by people who speak a different language.

Language differences have long been an obstacle to U.S.-based firms for information collection and transmission. Hofstede (1991) provides data on the ability of people in various countries to converse in foreign languages. In the U.S. and U.K., over 75% of the local population are unable to speak in even a single foreign language. In contrast, in the Netherlands, 75% of the people speak at least one and 30% speak two languages in addition to their native tongue. At least one foreign language is spoken by a large percentage of the people in Germany (40%), France (33%) and Spain (32%).

An important question is into which, and how many languages should the software be localized? Upon reflection, one realizes that the steps taken to localize to one additional language must be retaken for each subsequent language. Selecting just four languages results in four times four, or sixteen one-to-one localizations. A deceptively simple solution emerges from a comparison of the number of native speakers in Table 1.

Table 1: Language Distribution

Language	Number of Speakers (Millions)
Mandarin	864
English	443
Hindi	352
Spanish	341
Russian	293
Arabic	197
Bengali	184
Malay	142
Japanese	125
French	121
German	118

Upon further reflection, the multidimensionality of the problems takes form. Many languages are not standardized: U.S., U.K. and Australian English have trashcan, wastebasket and rubbish can, respectively. In this short list in Table 1, one sees four classes of language: simple, contextual, multi-byte, and directional. For example, Arabic is written right to left for text, but left to right for numbers (which, by the way are not the "Arabic" numbers used in many counties).

Moreover, English textstrings are usually shorter than equivalent phrases in other languages--as much as 50% shorter than equivalent German phrases. Data suggest the growth in string length during localization shown in Table 2.

Table 2: Textstring Length Growth

English Text Length	Growth
1-4 characters	100%
5-10	80%
11-20	60%
21-30	40%
31-50	20%
over 50	10%

Understanding the above differences will encourage developers to plan ahead to minimize re-sizing controls, menus, etc. String resources (strings, bit maps, icons) should be placed in a separate file. This increases efficiency, security (only necessary to outsource the localization), effectiveness (will not overlook anything).

Cultural Localization

What is the relevance of culture for developers of information systems? Culture operates subtly, often on the semiconscious level. It has been aptly compared to a computer program that once activated by commands or stimuli begins to operate automatically and in a seemingly independent manner. Nelson and Clark (1994) suggest that *any* technology can be considered as a culturally embedded, value-laden activity by which humans transform neutral reality to practical ends with tools and procedures. They further suggest that culture can be viewed as an independent external variable, that goes beyond the influence of organizational context that is introduced through membership, and revealed in patterns, actions and attitudes of individual members. Alternatively, culture can be a moderator variable, which may strengthen or weaken relationships among variables of interest. Finally, culture is not a power, something to which events or processes can be causally attributed: it is a context, something within which they can be thickly described.

Shore and Venkatachalam(1995) suggest that culture is an important variable, but that its influence may be indirect, and difficult to isolate or measure. They suggest that culture may play a significant role in three of four information systems development life cycle stages: requirements, high level design, and implementation/evaluation. The fourth stage, low level design, is relatively unaffected by cultural differences due to standards as described in the preceding two localization categories. Their framework draws attention to the issue of culture at the beginning of the development cycle and may reduce failures and delays.

Kendall (1996) defines a metaphor as a readily available shorthand symbol of corporate culture or subculture, and suggest that metaphors possess attached ideas that resonate when the metaphor is mentioned. As an example of the foolishness of allowing designers to continue to rely on metaphors that may not work in other cultures consider that the metaphor for chaos varies as: zoo (U.S.), fish market (S.E. Asia), brothel (France), and a circus (Germany). She suggests shared metaphors such as the Olympics.

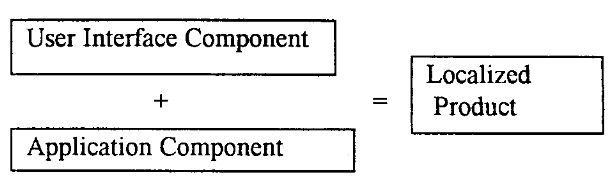
Developers should consider the following suggestions from Fernandes (1995), who contends that culture has four components that may affect interface design: values at the core, surrounded by rituals that reinforce, the heroes that are part of the rituals, and symbols.

- Understand your cultural blind spots--people are ethnocentric--blinded by their own culture.
- Where possible, transcend the components of culture by emphasis on common experiences, history, geography, science.
- Remove taboos such as numbers, human body parts, animals, food, religion.
- Present culturally appropriate aesthetics by studying the local history of design.

Internalization

The process of localization is generalized by a process termed *internationalization*, which is the proactive enabling of a software product so that it may be localized. The process involves creating a base design that can be changed for various markets. It provides a least common denominator functionality. A proposed development model for designing global information systems is shown in Figure 1. The model suggests, as does Kano (1995) and Luong (1995) that language-specific localization concerns, such as the user interface be addressed during the design phase of software development.

Figure 1: Global Systems Development Model



Ives and Jarvenpaa (1991) distilled from interviews four common approaches for managing global information systems. The first approach favored subsidiaries operating under their own information system initiatives, with the resulting systems built using foreign, national standards and local vendors. Consequently, subsidiary applications are customized to local needs and bear no resemblance to centralized headquarters' developed applications. In direct contrast, a second alternative approach is headquarters-driven global development with corporate-wide decisions imposed on all subsidiaries. A third approach, labeled intellectual cooperation, has the salient characteristic of joint application development targeted at achieving innovation and flexibility. Finally, integrated global development requires a planned common architecture in order to be simultaneously efficient and responsive in global markets. Table 3 portrays the software development options that correspond to these four strategic approaches for U.S. firms when entering target global markets.

Table 3: Development Options

Develop	Internationalize	Localize	Manufacture
Target	Target	Target	Target
U.S.	US.	U.S.	U.S.
U.S.	U.S.	Target	Target
Both	Both	Target	Target

Moving beyond the software engineering issues to application-specific concerns suggests the magnitude of the localization challenge. Consider localized version of a human resource management application for a multinational firm. In the U.S. it is illegal to track employees' religion, but in Germany it is required. In Ireland you must track religion but cannot associate it with an individual. Other examples of application-specific concerns that impact localization efforts include gasoline, heating, and food allowances.

Watson, et al. (1994) observed that the designs (and tests) of current group support systems (defined as an integrated combination of computer communication and decisions support technologies designed to support group work) are based on North American concepts of desirable group behavior. Until recently, exporting software to suit local customs, laws or convention focused on altering the technical facilities, such as dictionaries. Cultural specific assumptions in group support systems are: importance of equal opportunity for each group member and preference for open and direct communication to resolve conflict. By comparing the U.S. and Singapore on Hofstede individualism and power distance scales, they concluded that culture will shape the adoption of technology—groups use technology to achieve objectives. A common feature of problem solving across cultures is information exchange, but anonymity should be a switchable feature of the application package.

Conclusion

The practice of international systems development has clearly run ahead, without pausing to wait, for information systems researchers and educators to provide either a theoretical foundation or curricula. Practitioners have discovered that growing global software demands that you be local first, then global. As in every aspect of global enterprise, the strategies for information systems development vary widely. Many firms refuse to localize their products, instead arguing that the U.S. provides 75% of the global supply of packaged software (see Carmel, 1996) or that 89% of Internet addresses are located in just four English-speaking countries (see Brake, 1996).

Cheney and Kasper (1993) use the information technology differences between Europe and North America to argue for the importance of developing an international IS curriculum. They insist that technically oriented information systems programs must respond to global shifts that require a working knowledge of diverse practices and standards in world markets.

For information systems educators, language localization concerns, such as the user interface can be addressed during systems design courses. Locale specific aspects of the application component should then be addressed during implementation courses and workshops that require programming proficiency.

Inherent in preceding discussion is the concept of a process. Although usability has emerged as a key factor in global software, design methodologies have remained unchanged. A the present time, the global software development process involves domestic creation of the product as soon as possible, modify as needed to enable localization and then, deliver the product to a localizer in the target market and hope for the best. The process should begin with identification of target cultures, development of a global base product, localize, and test. Kumar and Bjorn-Anderson (1990) explain that methodologies have built-in value biases reflecting the value priorities of the culture in which they were developed. If overtly espousing alien values, will not be accepted.

References

Brake, D. (1996). "The U.S. Wide Web," *Salon* (Apple On-line), 30(September):3-6.

Carmel, E. (1996). "American Hegemony In Packaged Software Trade And The 'Culture Of Software'," *The Information Society*, 12(4).

Cheney, P.H., and Kasper, G.M. (1993). "Responding to World Competition: Developing the Global IS Professional," *Journal of Global Information Management*, 1(1):21-31.

Deans, P.C. (1992) "Information Systems Curriculum Development: Incorporating the International Dimension." *Information Resource Management Association Workshop.*

Fernandes, T. (1995). *Global Interface Design: A Guide to Designing International User Interfaces.* Boston: AP Professional

Frye, C. (1997). "Peoplesoft Goes Global," *Software Magazine*, (March):62-66.

Granger, M.J., D.L. Schroeder, B. Rollier, L. Esnault. (1992). "A Framework for Internationalization of the MIS Curriculum." In M. Khosrowpour (Ed.) *Proceedings of Information Resources Management Association International Conference*, 257-261.

Ives, B. and Jarvenpaa, S.L (1991). "Applications of Global Information Technology: Key Issues for Management." *MIS Quarterly* (March): 32-49.

Kano, N. (1995). *Developing International Software for Windows 95 and Windows NT*, Redmond, Washington: Microsoft Press

Kendall, E.J. (1996). "Global Cooperation: Using Metaphors to Communicate and Avoid Corporate Culture Clashes." *Decision Line* (Sept/Oct):4-6.

Kumar, K. and Bjorn-Anderson, N. (1990) "A Cross-Cultural Comparison Of IS Designer Values," *Communications of the ACM*, 2(3):5-12.

Luong, T.V., Lok, J.S.H., Taylor, D.J., and Driscoll, K. (1995). *Internationalization: Developing Software for Global Markets.* New York:John Wiley & Sons.

Nelson, K. and Clark, T. (1994) "Cross-Cultural Issues in Information Systems Research," *Journal Of Global Information Management*, 2(4):19-27.

Palvia, S. (1992). "Introducing Global IT Issues into the Business School Curriculum." *Information Resource Management Association Panel Discussion.* (May)

Shore, B. and Venkatachalam, A.R (1995). "The Role Of National Culture in Systems Analysis And Design," *Journal Of Global Information Management*, 3(3): 5-14.

Uren, E., Howard, R. and Perinotti, T. (1993). *Software Internationalization and Localization: An Introduction.* New York: Van Nostrand Reinhold.

Watson, R.T., Ho, T.H., and Raman, K.S. (1994). "Culture: A Fourth Dimension Of Group Support Systems." *Communications of the ACM*, 37(10):45-55.

Software Submission using Client/Server Architecture In a Windows Network Environment

Luann Stemler
David Doss
Bill Swafford

Applied Computer Science Department
Illinois State University
Normal, Illinois 61790-5150

(309) 438-3228

lstemle@rs6000.cmp.ilstu.edu
dldoss.@rs6000.cmp.ilstu.edu
beswaff@rs6000.cmp.ilstu.edu

Abstract:

This paper describes a system for program submission using the client/server model. A Visual Basic front-end is combined with a C client to provide access to a C server.

Introduction

In the fall of 1994, the curriculum of our department changed from a PL/1 mainframe based programming environment to one using C on microcomputer development systems. In addition to the changes in language and platform, a weekly two hour closed laboratory component was added to the introductory programming course. Previously in programming courses, faculty had been able to easily control mainframe test data sets so that grading student programs could be carried out by carefully checking program listings together with program output. With the implementation of the new curriculum, student programming assignments had to be submitted to faculty in the form of both a program listing and a floppy diskette containing the program source code.

The introductory programming course is organized into forty students per lecture section with two twenty-student laboratories per lecture. There are typically six programming assignments per lecture section. After the first two weeks of each semester, students are required to write from one to three small programs per week as a part of the closed laboratory experience. These activities produce approximately 30 programs per student, per semester, for a grand total of 1200 programs. Of course, most of these programs are small and easy to grade, but the overhead of grading either twenty laboratory or forty lecture programs that reside on separate diskettes is considerable. A non-trivial amount of time is consumed in inserting and removing diskettes, as well as in accessing the program files that reside on the diskettes.

In addition to the time required in manipulating so many diskettes over the course of the semester, the instructor must invest additional time in order to maintain the connection between a students program listing and the proper diskette containing the program source code. Although it is true that some of the laboratory programming activities are short enough that they could be adequately graded without a program listing, none of the lecture programming assignments fall into this category. In order to keep their program listing and diskette together, the students were required to submit their assignment materials in a folder or envelope. While this solution maintains the required connection between listing and source, it does add even more overhead to the grading process.

The process described above proved to be workable, but was so tedious that a better solution was sought. Since the faculty had little experience with PC based networks, the obvious solution did not present itself immediately. However, by the second year of the new curriculum, a program had been developed and placed on all departmental computers that would allow students to submit their source code directly from their diskette in the a: drive to a network server. The server was organized by course and section, so that programs for a given class were all collected together.

Visual Basic was chosen as the tool with which to develop the Submit program because of the GUI front end it provides. The faculty member who actually wrote the program already had expertise in using this development environment, so there was no initial learning curve. However, new facilities incorporated into the tool by Microsoft have led to improved usability characteristics of the Submit program.

Design of GUI Interface

The first problem in the program design was to determine a methodology for saving the student's files on the server. It was decided that each course would have a directory on the file server with each section of the course having a subdirectory within the course number's

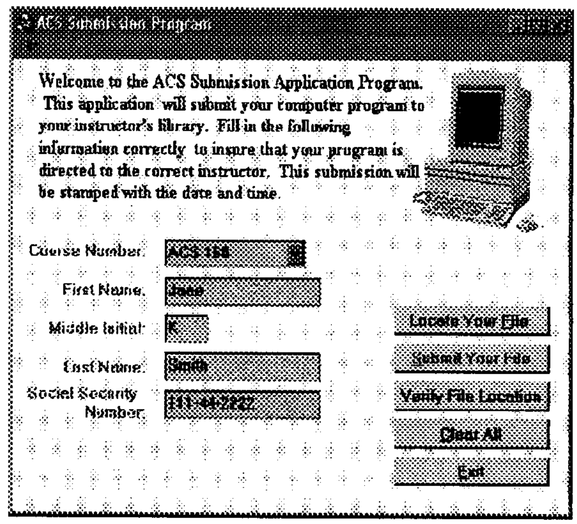


Figure 1

directory, for example ACS 168, Section 01 would have its own directory path of **T:\ACS168\01**. The student must click on the arrow in the combo box adjacent to the course number and select the course title. This action brings up a new visual display screen. The program reads a database stored on the server for the course section information. The section numbers are displayed in a list box on the left side of the form, as shown in *Figure 2*. The student must click on their section number to display the correct information regarding instructor and time of the course. This information needed to be displayed because many students could not remember what section they were in, but could usually remember the

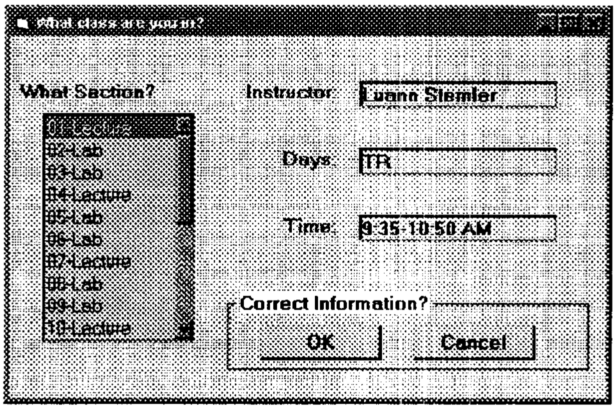


Figure 2

instructor's name and time of the course.
A database was chosen to store the course information due to the quantity of maintenance needed at the beginning of each semester and the ease of manipulating the data. Visual Basic Professional edition has a built-in database that was used in this project for convenience. When the selection of the course number and section number is complete, the path to the correct server subdirectory is determined. However, the path is not displayed for the student. Each instructor has access to the subdirectory of their corresponding section. The original design was concerned with servicing only the single multi-section course, but the solution has proven to be flexible enough to be used by any number of courses. Currently there are 10 different courses using the program with each course having multiple sections.

The next methodology to be determined was how to store all student files enrolled in the same section in a subdirectory with unique names. The convention that was previously used on the mainframe for student program names was adopted. Each student's file is named using the first letter of their first, middle, and last name along with the last four digits of their social security number, plus a single digit project number. The project number was added to give students the ability to save multiple programs in an instructor's directory. For example the file name for the information collected in *Figure 3* would be JKS22222.C.

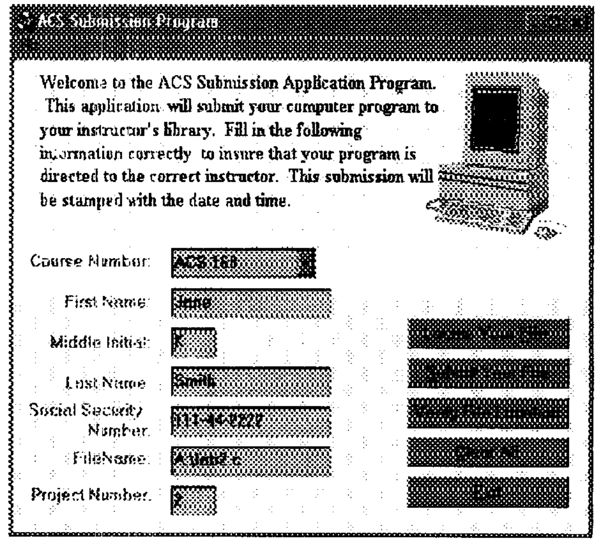


Figure 3

The student must use the ***Locate Your File*** button (*Figure 3*) to select the file they wish to submit. A common dialog box will appear in which student identifies the file to be submitted. The student then enters the project number and clicks on the ***Submit Your File*** button (*Figure 3*). The date and time of the server is used for the program when a student file is copied to their instructor's directory. A program can be submitted any number of times, and only one copy is saved, that being the most recent submission.

Most students were uncomfortable about relying on blind faith that the file was stored in the correct instructor's directory. An addition button was included, ***Verify File Location***, that allowed students to verify their file was copied to the correct location. The directory location is again invisible to the student. The student will only see the filenames that have their own initials and last four digits of their social security number, as shown in *Figure 4*. Files are copied with the original file extension because of the various courses that are using this Submit program.

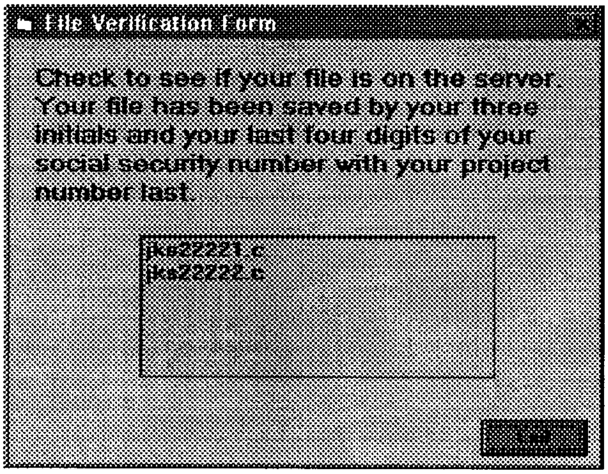


Figure 4

Improved Version

The first version of the Visual Basic Submit program was used successfully for a full academic year and it greatly improved the program handling process for the instructors. However, two observed potential problems led to the consideration of further changes. The first possible problem is related to the fact that student programs are collected to hidden subdirectories on a network server to which the students have write and create only access. Using the *dir* command with the *ah* switches, hidden file and directory names can be listed. Once the name of a hidden directory is known, the *dir* command can be used to list the files in the hidden directory. Then it would be a simple matter for someone with write and create access to systematically destroy all the student files. The existence of virus programs strongly suggests that someone might do just that.

The second possible problem is related to the use of the date/time stamp associated with submitted programs. This stamp is typically used to determine if a program was submitted by the assigned deadline, and if not, how many penalty points to assess. In the initial system, this date/time stamp is generated by using the date/time values of the computer from which the student submits his/her program. It is a very simple matter to reset these values backward so that a late program appears to have met the deadline requirements. This is not as malignant an activity as destroying other student programs, but is much more likely to happen.

The solution to both of these problems is to convert the existing Submit program into a client/server application. The server will save programs on a network drive to which the students have only the access provided by the server. This access will consist of saving a program submitted by a student to the appropriate subdirectory, and providing a student with a directory listing of programs that he/she has previously submitted. The first problem is defeated because students cannot use the *dir* command on the server drive since they have no direct access to it. The date/time problem no longer exists because the server uses the date/time values of the computer on which it is running, and not those of the client computer.

Converting to the client/server model allows the use of the existing Visual Basic Submit program to continue to provide the GUI interface to the students. The necessary network protocol sequence to access the services of the server is added by using a Dynamic Link Library (DLL) that is called by the Visual Basic module once all student information is collected. The DLL actually read the file from the student's diskette using information collected by the Visual Basic program, and sends it to the server.

An added benefit of the client/server version of the Submit program is that students are now able to submit their programs remotely using their own computers through a modem to access the university network. Students see this as a great convenience.

Conclusion

The Visual Basic module is written in version 4.0, so it can be run on both Win NT/Win 95 (32-bit) and Win 3.x (16-bit) systems. The server is written in Visual C++, version 4.0, and can run on Win NT and Win 95 systems. Because there are still a large number of Win 3.x systems in use, there are actually two versions of the client DLL module, one in Visual C++, version 4.0, for 32-bit systems, and one in Visual C++, version 1.52, for 16-bit systems. This software will be made available to interested parties from our departmental Web site.

It is too early to determine how gracefully the current version of the Submit software will handle the necessary volume of student program submissions. But the multitasking capabilities of the 32-bit operating systems virtually guarantee that a version can be developed that will handle the required load.

Bibliography

Cooper, Alan (1995). *About Face The Essentials of User Interface Design*. Foster City: IDG Books Worldwide, Inc.

Hall, F., Towfiq, M., Arnold, G., Treadwell, D., & Sanders, H. (1993, January 20). *Windows Sockets: An Open Interface for Network Programming under Microsoft Windows, Version 1.1.* Available: ftp://ftp.microdyne.com/pub/winsock.

Microsoft (1995). *Microsoft Visual Basic Programmer's Guide*. Redmond: Microsoft Press.

Microsoft (1995). *Notes for Developing DLLs for use with Microsoft Visual Basic Version 4.00 (VB4DLL.TXT)*. Available: ftp://ftp.microsoft.com.

Microsoft Corporation (1995). *Visual C++ Microsoft Foundation Class Library*. Redmond: Microsoft Press.

Scott, C., Shannon, B., Font, F., & Hatfield, B. (1995). *Visual Basic 4 Unleashed*. Indianapolis: Sams Publishing.

Webb, J., McKelvy, M., Martinsen, R., Maxwell, T., & Regelski, M. (1995). *Special Edition Using Visual Basic 4*, Indianapolis: Que.

COMMON GATEWAY INTERFACE (CGI) PROGRAMMING FUNDAMENTALS
A PRE-CONFERENCE WORKSHOP DESIGNED FOR EDUCATORS

PRESENTED BY

DR. NARAYAN MURTHY AND DR. STUART A. VARDEN
PACE UNIVERSITY
PLEASANTVILLE, NEW YORK 10570

INTERNET ADDRESSES: murthy@cs.pace.edu & vardenf@pace.edu

GENERAL DESCRIPTION

Most Web pages are still static, and offer no provisions for interaction between the client and server. This half-day workshop is designed to provide educators with the basics of CGI programming in support of interactive services via the World Wide Web.

Participates are expected to be familiar with the elements of Hypertext Markup Language (HTML), but need not be "experts."

TOPICS TO BE COVERED

Topics to be covered in the workshop include:

* HTML Forms

* CGI Programming

  - Relationship between HTML form and CGI program
  - Comparison of GET Versus POST
  - Parsing query data
    . GET Versus POST
    . Environment variables
  - Generating HTTP headers
  - Generating HTML code
  - CGI tools and examples
  - Troubleshooting CGI programs

* Presentation of Applications

  - E-mail feedback
  - Remote appointment/calendar management
  - On-line test administration

INSTRUCTIONAL ENVIRONMENT

The workshop will be delivered by means of lecture, discussion and demonstration.

Using The Internet To Teach Classes

Panel Session

Dr. George Fowler, Texas A&M University
Dr. Wayne Headrick, New Mexico State University
Dr. Ahmed Shabana, Texas A&M University
Representative from ZDNet University

The Internet has become a knowledge tool available to most teachers and students. It has been used for researching topics of interest, for helping with assignments, for studying other cultures, for communicating with others all over the world and for entertainment. This Panel will discuss the relatively new use of the Internet for teaching classes.

The Internet may be used to augment the class which still meets formally with the instructor. It may also be used in a more distance learning scenario in which students are presented the entire course over the net. This technique does not require the student to be in a formal classroom setting that meets on a regular schedule.

The Panelist will discuss these aspects of using the Internet to teach. They come from a variety of backgrounds and experiences. Three have extensive classroom experience with the internet and the ZDNet representative will add the commercial perspective that offers courses for continued education.

Tentative topics of discussion are:

- Which classes best fit the model for the internet?
- What materials are best utilized by students from the net?
- What hardware/software requirements are placed on the University or College?
- Reasons for and against off-campus courses
- Reasons for and against off campus degree programs
- Impact on faculty time
- Impact on faculty student evaluations

Author, Presenter, and Panelist Index