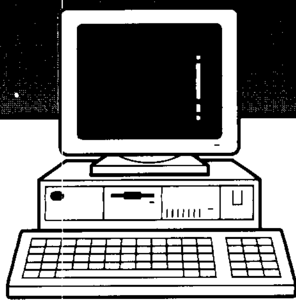
ISECON '94

Information Systems Education Conference



IS Education

For Today and for the Next Millennia

October 28-30, 1994

The Galt House Hotel

Louisville, Kentucky

Proceedings of the Eleventh Information Systems Education Conference

Conference Proceedings

SPONSORED BY
DPMA EDUCATION FOUNDATION

University of Houston-Downtown
Finance, Accounting and Computer Information Systems

WELCOME TO ISECON '94

IS EDUCATION FOR TODAY AND FOR THE NEXT MILLENNIA

I am very pleased to welcome you to the 1994 ISECON conference. On behalf of the Board of Regents of the Education Foundation of DPMA, the Steering Committee, and the Track Chairs, I would like to thank you for your attendance and participation in ISECON '94.

As we all know, our society is now living in the Information Age. As academics who teach or administer academic programs in this Information Age we know the difficulty in trying to "keep up" with the many advances that come our way. It is our hope that the valuable time that you spend at this conference will enable you to add to your "keeping up" skills.

ISECON '94 is a gathering of academics that will hopefully absorb and pass on knowledge about information systems to students. You, the participant, get to share your experiences with others. As I have stated elsewhere, it is always assumed that the mark of a successful conference, by the attendees, is that if each attendee leaves the conference with at least one new idea or presentation then the person felt it was a beneficial conference--we hope that you will have that experience.

Again, welcome to ISECON '94.

Sincerely,

Herbert F. Rebhun, Ph.D.
Conference Chair

One Main Street, Houston, Texas 77002

DPMA

Association Of Information Systems Professionals
EDUCATION FOUNDATION

Dear Information Systems Educators

On behalf of the DPMA Education Foundation Board of Regents, it gives me great pleasure to welcome you to ISECON '94 here in Louisville. This is a excellent opportunity to learn from and share resources with your peers.

The Conference Committee, under the chairmanship of Dr. Hebert Rebhun, has put together a great conference for you, which allows you to share in the latest thinking by the nation's leading educators. The education programs are among the best available anywhere and we encourage you to take full advantage of them.

While you are in Louisville, I hope you will take advantage of the many sights and sounds which make this city a joy to all visitors. What else can be said about historic Churchill Downs, the home of the derby. Or consider a leisurely cruise on the Ohio river aboard the " Belle of Louisville". There is plenty to do for all.

Be sure to attend the Sunday luncheon at which we recognize the DISEA winner. This award, sponsored jointly by the DPMA Education Foundation and the DPMA EDSIG, annually recognizes the educator of the year. This event is a highlight of the conference, since the award recipient always has an interesting presentation.

All in all, the package provides a fantastic educational experience. The sessions are current and global in nature, with an outstanding keynote speaker, and a city that takes its hospitality seriously. The only thing left to do is for you to participate.

Enjoy and Welcome to ISECON '94.

Best Regards

Leo C J de Folter, CMA
President,
DPMA Education Foundation
Board of Regents.

505 Busse Highway, Park Ridge, IL 60068-3191 - (708) 825-8124 - Fax (708) 825-1693

I S E C O N ' 9 4

S T E E R I N G   C O M M I T T E E

DPMA EDUCATION FOUNDATION PRESIDENT
Leo C. J. deFolter, CMA
Winnepeg, Manitoba
Canada

CONFERENCE CHAIR
Dr. Herbert F. Rebhun
University of Houston- Downtown
Houston, Texas

PROGRAM CHAIR
Dr. Eli Cohen
Wichita State University
Wichita, Kansas

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

EXHIBITS CHAIR
Dr. Mary Sumner
Southern Illinois University - Edwardsville
Edwardsville, Illinois

PLACEMENT CHAIR
Dr. James E. Trumbly
University of Texas - El Paso
El Paso, Texas

LOCAL ARRANGEMENTS CHAIR
Mr. Ron Booth
Sullivan College
Louisville, Kentucky

EX-OFFICIO, ISECON '93 CHAIR
Dr. George Fowler
Texas A & M University
College Station, Texas

I S E C O N ' 9 4

P R O G R A M T R A C K C H A I R S

Information Systems Curricula
Dr. Jon Clark
Colorado State University

Small/Community Programs and Industry & Education Relationships
Dr. Larry W. Cornwall
Bradley University

International Information Systems Issues
Dr. Oscar Flores
Universidad Autonoma de Nuevol Leon

Computer Information Systems Program Directions and Issues
Dr. Karen Forcht
James Madison University

Emerging and Leading Edge Technologies
Dr. Sorel Reisman
California State, Fullerton

ISECON '94

PROGRAM REVIEWERS

Macedonio Alanis-Gonzalez
State of Nuevo Leon

Greg Alexander
Davenport College - Lansing

Longy O. Anyanwu
Georgia Southwestern college

Irv Englander
Bentley College

Karen Forcht
James Madison University

Charles W. Koop
DeVry Institute of Technology

William L. Lomerson
Hardin Simmons Univesity

Kathryn McCubbin
Christopher Newport University

Joan K. Pierson
James Madison University

James B. Pick
University of Redlands

Jose M. Sanchez
Instituto Tecnologico Estudios Superiores de Monterrey

John F. Schrage
Southern Illinois University - Edwardsville

Bruce A. White
Dakota State University

I S E C O N ' 9 4

T A B L E   O F   C O N T E N T S

page

FRIDAY, OCTOBER 28

SATURDAY, OCTOBER 29

PROGRAM SESSION   10:00 - 11:15 a.m.

PROGRAM SESSION 1:00 - 2:15 p.m.

PROGRAM SESSION 2:45 - 4:00 p.m.

SUNDAY, OCTOBER 30

PROGRAM SESSION 8:30 - 9:45 a.m.

Development and Review of IS'95-A Joint Curriculum of DPMA, ICIS/AIS and ACM for Four Year IS Programs

Workshop

Panel: Herbert E. Longenecker, Jr., Chair
Jon D. Clark
J. Daniel Couger
David L. Feinstein
John T. Gorgone

Both the DPMA (1981, 1986, 1990) and ACM (1982) have been active in curriculum development for programs of Information Systems, and in the past have produced separate models. According to survey information these models are the basis for most four year programs of information systems for programs in business schools (~50%) and for non-business programs. The leadership in DPMA, ICIS/AIS and the ACM have been working to bring together all parties together to develop a unified curriculum model based on the strengths of all who are involved.

The application of information technology to the development and management of organizational systems continues to grow in importance to the survival of and well being of current organizations. The IS curriculum needs to evolve to ensure that students are not only exposed but proficient with the use and application of technology based development, and to the thought processes needed to develop the intellect to cooperate successfully in our rapidly changing work-place in which principle centered behavior must ensure high quality to maintain competitive advantage.

This new model curriculum is based on several national surveys, and on the work of its joint committee. It is now appropriate for interested IS faculty to read, evaluate and take part in improving the documents. This panel will be conducted by four members of the executive committee of the joint task force to present the model, and initiate the review process. Copies of the curriculum document will be made available. The panel will present some of the underlying features of the curriculum, and solicit discussion on several questions thought to be of general interest:

1. How is the common body of IS knowledge going to change? What are the major design principles of an IS curriculum?

2. Is there only one way to build an IS curriculum? How can diverse needs be incorporated into a single model? How do the design principles allow integration of materials in your environment?

3. How will IS programs exist in schools of business? What modules can be integrated into which areas? What is the importance of IS program outside the school of business?

4. What cooperative role can computer science courses and departments play in the IS curriculum?

5. What is the role of IT vs Management in IS curriculum? How can concepts of quality, principle centered leadership and concepts of continuous improvement be integrated into different courses?

6. What are the relevance, demands, and mechanisms for accreditation of IS programs?

ENHANCING MIS EDUCATION WITH DPMA'S PROFESSIONAL DEVELOPMENT PROGRAM (PDP) AND INDUSTRY STRUCTURE MODEL (ISM)

PANELISTS

Daniel F. Kesselring
Manager, Professional Development Program
Data Processing Management Association (DPMA)

Melvin A. Franz
Director of Information Services
Central Missouri State University

Charles W. Koop
Curriculum Manager of Telecommunications Management and CIS
DeVry Institute

ABSTRACT

The Professional Development Program (PDP) was developed by the British Computer Society to provide an industry wide set of guidelines and standards for planning and tracking career development of IS professionals. The program's success in Europe was recognized by DPMA who licensed PDP for distribution in North America. The Industry Structure Model (ISM) and training plans which form the system's foundation identify performance standards for more than 200 IS positions.

The Industry Structure Model provides a particularly valuable reference point for evaluating a CIS/MIS curriculum. The ISM's classification of positions and skill levels identify essential tasks/attributes that can be compared with student outcome measures of undergraduate, graduate, and Continuing Education programs. Those programs involved in AACSB accreditation have completed student outcome/objective lists that are "ready made" to compare with the ISM's list of position attributes. The comparison aids in identifying needed job entry level skills and training that is missing (or weak) in a curriculum. PDP not only allows for curriculum and course analysis but also aids in the selection of electives to individualize programs for students interested in specialty areas.

This workshop will provide a detailed description of DPMA's Professional Development Program (PDP) and Industry Structure Model (ISM). It will demonstrate how educational institutions can effectively use PDP and ISM to develop a curriculum responsive to the needs of students as well as industry. Also, it will show how students and faculty advisors can use PDP and the ISM to better prepare students for real career opportunities in industry.

ISECON 1994
Half-Day Workshop: 4 hours
Using COBOL to teach Object-Oriented Design

Kathleen Hennessey, Ph.D.
Professor, Information Systems and Quantitative Sciences (ISQS), and
Director, Institute for Studies of Organizational Automation (ISOA),
College of Business Administration, Texas Tech University.
Lubbock, Texas 79409-2101.
(806) 742-1609, fax -3466 806) 742-3466 odakh@ttacs.ttu.edu.

PROGRAM

A "live" case study , along with a student's solution to the case study, will be demonstrated during the workshop.

1:30 p.m. Introduction
Part I: Teaching Object-Oriented Constructs for System Design

* polymorphism concept
* object attributes: aggregation, classification, association, classification
* object representation
* inheritance and state transitions of objects

2:15 p.m.
Part II: Teaching the Relationship between COBOL and Object-Oriented Constructs

* ANSI 85 COBOL
* invoking functions
* linking to data records: intrinsic objects
* ANSI standard OO COBOL

3:00 Break
3:15 p.m.
Part III: Teaching Object-Oriented System Design in the COBOL Environment

* defining real-world objects
* abstractions from subsidiary assertions
* grouping of objects
* ranges of values
* methods and object states

4:00 p.m.
Part IV: Using Microfocus COBOL as a case tool for system design and prototyping - a hands-on session

* specifying objects through screen design
* linking objects across screens
* generating file specifications
* generating and compiling COBOL code
* running the prototype
* object-oriented design with the Dialog System

Development of a Specialized Laboratory to Support the I/S Curriculum

Irvin S. Englander
Donald R. Chand
Computer Information Systems
Bentley College
175 Forest Street
Waltham MA 02154
ienglander@bentley.edu
dchand@bentley.edu

ABSTRACT

With student ownership of increasingly powerful personal computers and the sustained investments over the years by the universities in their general computing infrastructure, it may seem strange to suggest that today's Information Systems (IS) programs need a specialized computing laboratory to adequately prepare tomorrow's IS professionals. This paper describes the factors that led the Department of Computer Information Systems at Bentley College to embark on the concept of a specialized computing laboratory for the CIS programs. It outlines our plans and discusses the issues that we have encountered during the first phase of our implementation of a CIS lab at Bentley College.

1.0 Introduction

It is widely accepted by both administrators and faculty that computer science education requires specialized laboratories that are separate from the general computing facilities of the College. On the other hand, there is little precedent for information systems programs to have a laboratory facility of their own. It is generally assumed that the combination of personal computers and a campus wide information system is sufficient to fulfill the requirements of IS programs. This assumption is based on the reasoning that personal computers, student owned or provided in the general computing laboratory, are effective means for learning the use of standard productivity tools such as word processing and automated spreadsheets, for exploring the use of single person operating systems and file systems, and for learning the principles of programming. Campus wide communications systems are well suited for teaching the concepts of e-mail and powerful information retrieval methods, such as those found through the facilities of the Internet. The multiuser mainframe environment is good for teaching IS majors about large scale database design and development concepts and issues.

We, at Bentley College, found that even a rich set of computing facilities consisiting of student-owned portable personal computers, a general lab full of IBM compatible and Apple Mackintosh microcomputers, a campus-wide e-mail facility, and a pair of multiaccess mainframes with different operating environments is insufficient to support the needs of an up-to-date IS curriculum. After careful considerations of the alternatives, we concluded that our general computing facilities need to be complemented by a specialized IS laboratory.

The objective of this paper is to share the thinking that led to the development of a specialized IS laboratory at Bentley College, the approach we took, and our early experience with this lab. The paper is organized as follows. We first provide a thumb nail sketch of the orientation of Bentley College, its computing facilities and its IS programs. The purpose of this next section is to provide a context for our discussion, and also to convey our perception that the facilities and programs at Bentley College are typical of the situation at many colleges and universities. In the third section we outline the limitations of existing shared computing environments to effectively support the learning needs of IS students and the goals of IS programs. In the fourth section we describe the process that led to the change in the objectives of the specialized computing environment from a CASE lab to a CIS learning lab. In the fifth section we discuss how we addressed the hardware, software and staffing issues of initiating and operating a lab. In the sixth section we describe our plans for the lab. These plans indicate our view of the direction in which IS curricula will move in the nineties. The seventh section describes an evolving methodology for effective use of the lab. We conclude the paper with our reflections on our experiences thus far.

2.0 Computing at Bentley College

Bentley College today is a large institution specializing in business education. The college has recognized the importance of computer fluency in business education, and has for many years provided extensive and sophisticated computer resources to meet the needs of its student and faculty population. The college strategy developed several years ago also included extensive faculty training, implementation of a plan to equip all freshmen with personal computers, and a program to integrate the use of the computer as a productivity tool across the curriculum. This effort to insure that Bentley graduates would be computer fluent and able to utilize technology to solve business problems has been successful by all accounts.

Today all full-time undergraduate Bentley students own or lease a personal computer. In addition, general computing laboratories provide personal computer facilities for graduate students and for.part time undergraduates, as well as providing a facility for software that is too expensive or too limited in use to justify purchase by students for their own machines. The laboratories provide a student help desk, backed up by staff personnel and by the academic computer services organization.

The campus wide information system provides VAX/VMS facilities, e-mail, bulletin board facilities, Internet tools, and CD-ROM information retrieval facilities. A smaller VAX computer provides an Ultrix based system, used mostly for the teaching of C and Unix.

The campus is fully networked. The VAX computers, laboratory PCs, faculty and staff offices, dormitory rooms, library, and classrooms are all interconnected by a large Pathworks network, providing both terminal and server access to the VAX/VMS system from nearly everywhere on campus. Students and faculty can access the library and media resources and communicate with each other and the community outside Bentley College using their personal computers from their offices and dorm rooms. Personal computer skills, information access, and e-mail capability are an integral part of the teaching process at Bentley College.

3.0 Limitations of the Shared Computing Approach

Despite the extensive capabilities of the Bentley computing facilities, we found that there are specific needs of IS programs that are not adequately met by the shared general computing facilities. This is somewhat surprising, because the layered computing architecture was designed to meet both the general and specialized needs of the academic programs. For example, even though every Bentley student possesses a personal computer, the general computing laboroatory was built to support the discipline specific or course specific needs of all the academic programs. This was based on the idea that the commonly needed affordable application software will be purchased by the students to run on their own personal computers, and that students will use the general computing laboratory for specialized needs that can not be met on the students' individual computers. An example of commonly needed affordable software in the IS program is Turbo Pascal. Examples of specialized needs in the IS program are the use of Excelerator/IS in systems analysis and design courses and the use of an OOA tool and Smalltalk in the object-oriented course.

Unfortunately, the general purpose lab has failed to support the course specific computing needs of CIS students effectively. There are several reasons for this:

• Software tools of importance to the IS program, such as CASE and object oriented tools, are outside the realm of general college use. As a result, the academic computer services is unwilling to support such software due to its limited use, the expense in terms of time and personnel resources for supporting these limited use packages, and the additional expertise required. Furthermore, the student lab assistants are not qualified to help IS students when problems arise.

•The heavy use of the software tools required by IS students ties up the lab, and makes the facilities unavailable to other students with simpler needs.

• General lab facilities are inadequate for many IS program needs, since they are primarily intended for individual, rather than group, effort. The facilities are unsuitable for long term repository storage, as required for team projects.

Our experience of using the general computing lab to support Excelerator/IS for the past five years and the OOA tool in our objected oriented course last semester, as well as the use of the campus wide information system to support system management, security, and operating systems courses for more than five years led us to the following conclusions:.

• The campus wide information system and the computer support services are best geared to the needs of the general faculty and student body. We found that general computing needs are often at odds with the specialized needs of the IS program. For example, the use of an XWindow interface would significantly enhance the learning capabilities of IS students, but implementing this interface for the entire system would be expensive, difficult to support, and possibly *more* difficult for non-IS students and faculty to use. The alternative of supporting multiple interfaces is not feasible because it increases the complexity of the system unreasonably.

• IS students are best introduced to system management, networking, and security concepts by allowing them access to the system at the highest levels. This need is contrary to the requirement of maintaining a secure and stable system for use by the general college body.

• It is not practical for most general computer labs to provide specialized hardware such as multimedia facilities, nor specialized software, since its use is so limited, and its support requirements so stringent. It is our experience that the college computing laboratories do not meet the special needs of IS programs as well as might be expected.

4.0 The Concept of a Specialized I/S Computing Lab

The limitations described above led us to the concept of a specialized laboratory facility centered specifically upon the needs to support the learning and use of CASE tools in our system analysis and design courses. The offering of the object-oriented course, last spring, expanded the scope of our specialized lab to include facilities for other courses. In the 1993 spring semester, the students in our senior level capstone course were assigned to develop a master plan for a CASE lab to support the teaching of advanced application development technologies. This project concluded that the CIS department needed a specialized laboratory which the students viewed as an advanced learning facility to support all aspects of the CIS curriculum. In the area of CASE, it recommended that the specialized lab needs to be equipped with products that cover and integrate the entire system development life cycle. Other important observations made include:

• CIS majors need a learning lab where students can learn from their peers as well as from well trained lab staff.

• The CIS lab must have an area specifically set aside for tools research.

• Documentation on the products used in the lab should be readily available in the lab.

• Since the business world demands the ability to work effectively in a team environment, the CIS lab should foster the ability to do group work.

• Ideally, the lab setting should be as close to the "real world" as possible in the sense that it should be equipped to experience real world software development processes, like running Joint Applications Development (JAD) sessions.

These observations by the students led us to change our orientation from a state-of-the-art CASE lab to a specialized CIS computing environment. This concept is characterized by

• a variety of hardware, including the standard PC hardware of the type found in the general computing lab, but also to include specialized facilities such as multimedia, and specialized computer systems such as Sun workstations and Power PC based systems. The machines in the lab will be networked, and may run under network operating environments that are non-standard for the campus.

• software that is officially not supported by the academic computing service.

• IS program and course specific use that may require specialized expertise to support student learning.

Software is clearly an important disinguishing attribute of the CIS lab. Yet it is not the software that is the driving force for establishing this lab. The primary reaon for establishing the lab is the unique pedagogy and the learning environment. Some of the key attributes of the learning environment are

• Availability of on-site expert help
• Availability of appropriate documentation
• Security of student work and repository storage
• Availability of machines to work for large blocks of time
• Physical space designed for group effort
• Accessibility to the machines for experimentation at the operating system level

It should be noted that the students in the capstone course who developed the plan for the specialized computing environment called it the CIS learning lab. The use of the term "learning" in the title is significant. They wanted to call attention to a need that we as IS educators must appreciate: that students need expert help, especially when they are learning to use a new tool, are working in an unfamiliar environment, or when they run into problems.

The second attribute stems from an important goal of IS education to instill in the student the desire and capability to understand and use system documentation. To achieve this goal, we need to create an environment where the requisite documentation is readily available.

The issue of security of student work arose from the observation that in systems development work with CASE tools, group members need access to the work of other members which cannot be coordinated by exchanging disks. Thus, the lab environment needs to be configured in such a way that the safety of student work from one session to another can be guaranteed. Ideally, the work of a group should be placed in a repository where team members can use tools that they are most comfortable with yet work on the same document or project.

The issues of students being able to work for large blocks of time and for group physical space arise from the nature of IS program lab usage. The norm in the general computing lab is to limit the use of machines to an hour or so, depending on the demand for those machines. These arbitrary time limits are not conducive for IS work. Most general computing labs are also designed for single user access to individual machines.

5.0 Issues Central to Operating the CIS Lab

After detailed study, The IS lab opened for business in January of 1994. The central issues that had to be resolved were staffing, training, and support, hardware and architecture issues, software tools, and faculty committment. The resoultion of each of these issues is discussed in the following sections.

Staffing, Training, and Support

Ideally, the lab should be accessible twenty-four hours a day, seven days a week. Since we do not have the resources nor the will to even consider this notion of twenty-four hours operations, we have initiated the lab by opening it at fixed hours during the day and the evenings, and have also provided the option to call in on the weekends and make an appointment to work in the lab. The role of the department assistants and the learning support we are developing is described later in the section entitiled "An Approach for Operating the Lab".

Hardware Architecture

The initial configuration consists of six 486DX2/66 local bus computers. Four machines are equipped with 340 megabyte hard disks; the fifth machine came with 510 megabytes, and the sixth with 2 gigabytes. The large capacity hard drive was provided to allow the sixth machine to act as a server. The 510 megabyte machine is also equipped for multimedia. It includes a CD-ROM drive, a sound card, and a 17 inch SVGA monitor. The other five machines all provide 15 inch SVGA monitors.These monitors were selected instead of standard 14" monitors to improve the ability to interact with CASE visual tools. All six machines are equipped with 16 megabytes of memory and a SCSI port. The lab has a SCSI-based portable CD-ROM drive that can be connected to any machine for loading software or for additional multimedia capability. A portable tape drive is also available for backing up each system.

At the present time, each computer is connected in a stand-alone mode, since the room made available for the lab was in a corner of campus not yet accessible to the Pathworks network. Installation of the network is expected by early summer of 1994. At that time, facilities of the lab will be network accessible from faculty offices and from classrooms. This will allow faculty to demonstrate IS lab software tools in the classroom, and to develop materials and access student work from their own offices. In particular, it will allow the faculty to experiment with a planned Lotus Notes facility in a convenient manner.

At present lab printing is performed on a single HP 4M LaserJet printer, shared between the various computers. Since the network is not yet installed, a switchbox is used to control printing. When the network is installed, the printer will be available via network from each station.

Software

Each machine is presently set up with Visible Analyst for CASE work, Visual Basic, and Lotus Smart Suites, all running in a MS-DOS/Windows environment. Some of the machines also provide C++ and Smalltalk

Faculty Committment

The success of the lab is directly linked with involvement and leadership of the faculty that teach courses involving emerging applications development technologies. Without their commit- ment to continous learning and retraining the concept of an IS lab will fail. The faculty member who is teaching the systems analysis courses this semester is developing an approach, described later, that addresses the issues of teaching tools when the instructor and the department assistants are not adept users of the tool.

6.0 Immediate Plans for Further Lab Development

- In the area of system analysis and design, a major goal of the CIS program is to provide an in-depth integrated understanding and expertise in the CASE based system development environments. Our strategy is to provide student access to one or more CASE based system development environments that support group work, and which can be used in a variety of courses. Since this particular goal was the primary motivation for the IS lab, it is discussed in more detail in the next section.

- Groupware tools, particularly Lotus Notes, are assuming a growing role of importance in IS environments. The IS Laboratory will serve as an experimental Notes environment that can be shared and explored by IS faculty and students. Departmental faculty members have already performed much of the work necessary to implement Lotus Notes within the lab.

- Since the computers in the IS lab are under departmental control, they can be used for student lab exercises in systems management and security. Current plans include equipping two or three computers with Unix. Students in the appropriate classes will be given super-user status, so that they may experiment with the computer at system level. Lab experiments in Unix system management are expected to start in Spring semester 1995.

• Plans for using the lab for networking and communication exercises and for multimedia are presently less well defined, but nonetheless represent important goals for the next year or two.
Hardware and software purchaes are being made with these additional goals in mind. The department expects, as a fundamental principle, to keep the lab in sync with the state of the art for important trends in Information Technology.

• Present plans include the addition of alternative workstations. This decision has been delayed primarily to allow the department to determine the appropriate selection. Possibilities include Alpha based machines, Power PC based machines, and Sun workstations.

• A similar situation is present in the area of system software. At the very least, we expect to provide Unix as an alternative operating system on each of the current machines. Other possibilities include OS/2 and Chicago. This decision will be made on the basis of importance to the curriculum and availability of software.

7.0 An Approach for Operating a Learning Lab

Although the IS lab has been in existence for just two months, it is already evident that the lab will have a significant impact on the IS program. For the systems analysis and design courses we have developed a promising four step approach that utilizes the lab to teach state-of-the-art tools and to operate effectively as a learning lab in the sense described by the students in their report. The process consists of demonstration, coaching, assignment, and feedback database steps.

Approximately 100 students, enrolled in two systems analysis courses and one systems design course, are using this lab to learn the Visible Analyst Workbench (VAW) and use it to support their group projects. The two systems analysis classes are using the VAW to develop process and data models and for prototyping screens and reports. The design class will also develop structure charts and migrate the VAW repository to a PC based database management tool (Lotus Approach and Microsoft Access) to complete the development of the application from start to finish using a CASE tool and a database management package.

To introduce the students to the features and functionality of the tool, the instructor makes several demonstrations in class. For example, the first demonstration showed how to draw a context diagram, explode it into a systems diagram, enter details in the repository, run the syntax analyzer, export and import VAW work, and generate reports on the process model details. The second demonstration showed how to enter the business rules, draw an initial entity-relationship model, run the syntax analyzer, perform a normalization check, run the key analysis check and perform key synchronization.
After each demonstration, students are assigned textbook problems to perform the functions demonstrated in class. Department assistants are available in the lab and they serve as coaches. Students are required to submit their work with a list of problems and issues they encountered in doing the assignment. These student inputs are analyzed by the instructor to understand the difficulties new student encounter when learning a new tool with limited guidance, and more importantly, to develop a database of problems encountered and ways to deal with these problems. Although this process became quite lengthy because the instructor has no prior experience with the tool and recreating the student problems is a cumbersome task, we are convinced that this database will become a key learning and support tool in our teaching / learning process. Our goal is to make this database available to the students both as a learning tool and as a trouble shooting tool. The limited data we have collected has already brought out issues and problems with the tool that we believe would be hard for even a gifted quality assurance analyst to uncover in this short time frame.

The department assistants in the lab are graduate students who have very limited experience with the

CASE tool, but they know the methodologies well and are fluent with the operating environment. As coaches they help the student focus on the right problem and they also provide pointers to where the students can find an answer to their problem.

Although we have not yet collected any formal student input on the learning environment, the informal feedback suggests that students are happy with the learning systems that are being continuously improved.

In addition to our experience with the system analysis and design courses, a number of students have approached faculty on other projects of interest to them. At their own impetus, several undergraduate CIS students are working with a faculty member to develop materials for a course in graphics and graphical user interfaces. Presently, this group is experimenting with Visual Basic, but they will expand to Visual C++ and XWindows when they becom available, shortly. Other faculty are already preparing to use the lab to build hands-on networking exercises and to teach students to manage a UNIX system environment. The lab will provide a prototype and preparation site for new tools such as Lotus Notes.

The IS lab has created considerable excitement for both faculty and students in the CIS department. A number of undergraduate students have commented that the laboratory has provided a psychological boost for CIS majors. Several have volunteered to serve as support people in the lab. It has been suggested that the lab may help us to retain and increase the number of CIS majors at the college.

8.0 Conclusion

Information Systems departments are under pressure to teach state-of-the-art tools in IS programs and to prepare our graduates to learn new tools of the trade. Client/Server, Object-Oriented methods, and I-CASE are three new technologies that need to be included in the IS curriculum. System management, security, and data communications require more hands-on experience to fully convey the practical aspects of the material. Our initial experience with the IS lab indicates that the lab will have a significant impact on the design and execution of the IS program. The learning lab approach using the demonstration /coaching/assignment/feedback database process appears to be an effective way for IS programs to keep up with new technologies.

Factors affecting curriculum design for non-computing professionals: A case study

R.C.MacGregor
Department of Business Systems
University of Wollongong
Northfields Avenue
Wollongong
Australia 2522

R.S.Cocks
Wollongong Veterinary Hospital

ABSTRACT

This paper examines the effect of organisational size, type of work, length of time using computers and overall satisfaction with computer technology on curricula inclusions in the Australian Veterinary Industry. The responses of 298 veterinary practitioners were examined. Results suggest that type of work, length of time using computer technology and overall satisfaction/dissatisfaction significantly affect the rating of certain subtopics which may be introduced into a computing curriculum. Contrary to recent small business findings, results further suggest that lack of finance is not a major disincentive to computer purchase.

INTRODUCTION

With the development of more 'job-oriented' courses of study, there has been a growing call for the involvement of employer groups into the process of curriculum development (Wattenberger and Scaggs 1979, El-Khawas 1985). Indeed, in countries such as Australia, government has pressed for the strengthening of links with both industry and foreign learning institutions (Dawkins 1988a,b) as a means of ensuring relevance of the final tertiary product.

Over the last decade many studies have been carried out with the view to determine the effect of introducing computer technology into a work organisation (Turner & Karasek 1984, MacGregor & Clarke 1988, Sharp & Lewis 1992, Williams 1992, Bergeron et al 1992). The results of these studies have provided a variety of factors which must be addresses when computer technology is introduced into the work setting. More recently, studies have been carried out examining the implications of some of these factors with respect to computer training (Raymond 1988, Kahn & Robertson 1992, Stauffer 1992, Hedberg & Harper 1992).

Given the advocacy, both at a political and academic level, for the involvement of employer groups in the development of curricula, little is known about how such decisions might be affected by the nature of the organisation or the satisfaction of with the computer technology used within that organisation. This paper examines factors such as organisational size, major type of work, length of exposure to computer technology and satisfaction with that technology and the effect of these on choices made in the design of computing curricula for non-computing professionals. The case study used is the Australian Veterinary Profession.

A BRIEF OVERVIEW OF THE AUSTRALIAN VETERINARY PROFESSION

The Australian Veterinary Profession is basically subdivided into four distinct groups: small animal, large animal, government and academic. Practices range from single person operations to multi-vet hospitals. The veterinary profession, like most areas of medical science, has always attempted to remain 'current' both in professional practice and in its course offerings. Today, a variety of journals, not only provide the profession with new medical techniques, but valuable information on all facets of practice management. The use of computer technology within the day-to-day practice management has been one of the many topics 'aired' within journal offerings. While the use of computers within the veterinary practice is by no means a new phenomenon, studies (Waldhalm 1991, MacGregor & Cocks 1991, 1994) suggest that the profession appears to be reticent to embrace computing as an academic offering within an undergraduate course.

Table 1
Curriculum Subtopic Groups

Subtopics	**Brief Description**
Strucutre and Function of computer hardware	study of the major components of the microcomputer
Programming	problem solving and program development
Database/Spreadsheet	the use of databases and spreadsheets in practical business problems
Practice Analysis	analysis of the major functions and data in the veterinary practice
Information Analysis	data modelling techniques
Office Automation	the integration of microcomputers into the office environment
Business Accounting Systems	the design and use of accounting software
Computer Evaluation Techniques	techniques for testing and comparing potential hardware and software purchases
Accountancy	introductory principles of accounting
Finance	introduction to corporate valuation and financial markets
Marketing	market segments, buyer behaviour etc.
Business Law	partnerships, contracts, liabilities etc.
Statistics	descriptive and inferential statistics
Management Principles	goal determination, implementatioon etc.
Interpersonal Skills	verbal, written, formal and informal communication

DECISIONS ADOPTED PRIOR TO THE STUDY

There have been many models of computer satisfaction. Yap, Soh and Raman (1992) investigated 118 small businesses and found that user satisfaction is positively associated with consultant effectiveness, level of vendor support, sufficiency of financial resources and the level of user participation. Their results further indicated that a number of factors appeared to impinge upon the perception of vendor support, improvement of income and efficiency. These factors included:

- size of the organisation
- the experience of the organisation in terms of computing
- the type of work for which the computer was being utilised
- the type of work that is normally carried out by the organisation.

While many of these models are valid, Igbaria (1993) suggests that many do not, at present, have an adequate test model and thus results can often be misleading. As such this study focused on: size of the practice, type of practice, number of years utilising computer technology, and overall satisfaction with that technology.

As with models of user perception of computer technology, there have been many studies examining computer curricula requirements, both in Australia and South East Asia (Lo 1991, Ang & Lo 1991a,b, Ang 1992). While many of these studies have used upwards of 50 subtopics, Seeborg & Ma (1989) considered that when examining non-computing professionals, the number of subtopics should be reduced. As such the current study has utilised 15 subtopics (see table 1).

METHOD

A questionnaire was developed which sought information as to whether a computer was being used within the practice. Those respondents who indicated that they did not use computers within their practice were asked to give reasons why they chose not to use computers. Those respondents who did indicate the use of computers were asked the size of the practice, the type of work normally carried out (small animal, large animal, government or academic), the length of time they had been using computers and their view of its suitability to the practice environment. The questionnaire also asked respondents to rank each of the 15 subtopics (1 not important .. 5 very important) in terms of their perceived importance for newly graduating veterinary practitioners.

Questionnaires were distributed throughout Australia through the Australian Post Graduate Veterinary Foundation as an inclusion in their regular information packages.

ANALYSIS OF RESULTS

4205 questionnaires were distributed. Useful responses were obtained from 298 veterinary practitioners. While the response rate (7.1%) was below the normally anticipated 10% - 30% response rate for questionnaire sampling, it was felt that the size of the return sample warranted examination. Furthermore, it was considered that since the questionnaire was one part of a large information package, this may have lowered the return sample size. Table 2 provides the demographics of the responses.

Table 2
Demographics of Sample Group

Practices using computers	226
Practices not using computers	72

Number of Vets within the practice

	Practices without a computer	with a computer
1 person	14	38
2 people	24	53
3 people	12	38
4 people	9	32
5 people	4	15
5+ people	9	50

Table 3 indicates the major reasons given by respondents for not using computers in their practice. As can be seen more than half of the respondents felt that the computer would alter working conditions. Indeed, the only other substantially supported reason for non-computerisation was that of expense.

Table 3
Reasons for Non-computerisation

Too expensive	23
Wont do what I want	10
Fear of the computer	5
Working conditions will be changed	34

Neergaard (1992), in a study of Danish small businesses, found that the size of the company affected the decision to computerise or remain in a non-computer mode of operation. Based on the study of Neergaard, responses to non-computerisation were compared to the size of the

Table 4
Overall responses of the importance of computer courses at undergraduate level

Rate	Small Animal	Large Animal	Government	Academic
Very Important	158	16	43	23
Important	26	4	6	3
Not Important	14	1	0	4

Table 5
Relative Importance of Information Systems Subtopics in Undergraduate Courses

Topic	Small Animal mean	Small Animal rank	Large Animal mean	Large Animal rank	Government mean	Government rank	Academic mean	Academic rank
Structure/Function of computer hardware	1.85	13	2.95	12	3.02	8	2.91	13.5
Programming	1.81	14	2.1	15	2.43	14	2.64	15
Database/Spreadsheet	2.6	1.5	4.05	1	4.7	1	4.55	1
Practice Analysis	2.55	4	3.65	3.5	3.47	4	4.27	2
Information Analysis	2.59	3	3.65	2.5	4.3	2	4.18	3
Office Automation	2.28	8	2.55	14	3.11	7	3.27	9.5
Buss/Acct Systems	2.6	1.5	3.6	5	3.33	5	3.64	7
Evaluation Techniques	2.08	11	3.25	10	2.74	12	3.6	8
Accounting	2.39	5	3.3	9	3.0	9.5	3.0	11.5
Finance	2.17	9	3.05	11	2.98	11	3.0	11.5
Marketing	2.38	6	3.5	6	2.7	13	3.27	9.5
Business Law	1.58	15	2.7	13	2.0	15	2.91	13.5
Statistics	1.86	12	3.35	8	3.84	3	4.09	4
Mgt Principles	2.36	7	3.95	2	3.17	6	3.7	6
Interpersonal skills	2.11	10	3.45	7	3.0	9.5	3.8	5

Table 6
Courses for Undergraduates

Groups being compared	Correlation Coefficient
Small Animal - Large Animal	0.766*
Small Animal - Government	0.650
Small Animal - Academic	0.711*
Large Animal - Government	0.698*
Large Animal - Academic	0.848*
Government - Academic	0.829*

* significant ($p<0.01$)

practice. Results indicated that the size of the practice does not affect the reasons given for non-computerisation.

The response were examined to determine the overall opinion of practitioners concerning computer courses at an undergraduate level. Responses were grouped into three categories:

Category	**Criteria**
Those who thought some form of computer training was very important	At least one of the subtopics rated 5 - very important
Those who thought some form of computer training was quite important	At least one of the subtopics rated 4 - important
Those who thought some form of computer training was not important	No subtopic rated above 3 - little importance

Table 4 indicates overall responses.

Table 5 shows the ranking of subtopics by practitioners, divided into the four major work types.

A Spearman's Rank Order Correlation was used to compare the subtopic rankings by each group. Table 6 summarises the Spearman's Rank Order Correlations.

Responses were examined using a chi-square analysis to determine if the type of practice affected the rating of subtopics. Table 7 shows the rating of subtopics subdivided into practice type. For Brevity, only those subtopics which are statistically significant are displayed.

The responses were examined to determine if overall satisfaction with computer technology affected the ranking of subtopics. Table 8 shows the ranking of subtopics by practitioners, divided into two groups - those who indicated satisfaction with the computer technology used within their practices, and those who were not satisfied.

Finally responses were examined using chi-square analyses to determine if length of time using the computer in the practice or the size of the practice affected the ratings of the subtopics. The results suggest that the size of the practice has no effect on the choice or rating of subtopics. Table 9 indicates those subtopic ratings which appear to be affected by the number of years the practice has been computerised. For brevity, the only data displayed in tables 7 is that which is statistically significant.

DISCUSSION

While studies (Neergaard 1992, Yap et al 1992) suggest that a major disincentive to computer purchase is the lack of finance, results from table 3 would suggest that at least in the Australian Veterinary Profession major concerns (67%) are other than financial.

An examination of the overall response to computer education at an undergraduate level (table 4) suggests that there is a perception by the practising vet for the need for computer studies within the overall undergraduate course structure. An examination of table 6 would suggest that with the exception of small animal and government veterinary professionals there is a high degree of consensus as to the relative importance of potential course subtopics.

Responses to subtopic rating were compare to the size of the practice, the type of practice, number of years using a computer and the overall satisfaction with the computer technology. While Neergaard found that the size of the organisation had direct bearing upon satisfaction, this was not apparent in the veterinary profession, nor did it affect the subtopic ratings. An examination of table 7, however, suggests that the type of practice does significantly affect rating of subtopics. While large animal, government and academic veterinary practitioners considered programming of little importance, the small animal vets were more equivocal. Database/Spreadsheets were rated lower by the large animal vets than any of their counterparts. In the area of Business and Accounting Systems, Accounting and Marketing there were distinct differences between the small and large animal groups and the government and academic groups, the former rating these important, the latter reducing their importance. This trend was reversed for the area of statistics.

An examination of table 8 suggests that there is a realisation by vets that for successful computer implementation to occur, practice analysis is essential. The data presented in table 8 also seems to support the findings of Yap et al (1992) vis., the need to address finance questions when computerising a practice.

When examining the data concerning length of time using computers, those practices which indicated that they had been using a computer for 3 years or 5 years appear to rate database/spreadsheet and statistics higher than their counterparts. There appears to explanation for this from the data in the current study.

Table 7
Ranking of subtopics by practice types

Subtopic		**1**	**2**	**3**	**4**	**5**
Programming **	Small Animal	45	39	41	33	32
	Large Animal	9	2	4	4	1
	Government	11	17	11	4	4
	Academic	4	0	5	0	2
Database/Spreadsheet **	Small Animal	5	11	27	52	94
	Large Animal	2	0	2	9	7
	Government	0	0	2	9	35
	Academic	0	0	2	1	8
Office Automation *	Small Animal	10	24	55	49	52
	Large Animal	3	5	5	2	5
	Government	3	9	15	16	2
	Academic	1	1	5	2	2
Buss/Acct Systems **	Small Animal	6	12	36	53	83
	Large Animal	1	2	6	5	6
	Government	2	7	19	9	8
	Academic	1	1	3	2	4
Accounting ***	Small Animal	9	15	52	59	56
	Large Animal	2	2	5	6	5
	Government	3	13	14	13	2
	Academic	3	2	1	2	3
Marketing **	Small Animal	12	18	44	55	62
	Large Animal	1	3	5	4	7
	Government	8	13	10	10	3
	Academic	1	2	3	3	2
Statistics ***	Small Animal	33	39	51	40	26
	Large Animal	1	2	8	2	6
	Government	1	4	13	10	17
	Academic	1	0	2	2	6

* significant ($p<0.1$) ** significant ($p<0.05$) *** significant ($p<0.01$)

Table 8

Ranking of the Subtopics by practitioners (overall satisfaction with computer technology)
Note:Only significant values have been printed

Subtopic		1	2	3	4	5
Practice Analysis *	satisfied	2	11	19	17	37
	dissatisfied	5	3	25	25	38
Accounting *	satisfied	4	13	17	33	20
	dissatisfied	9	15	29	19	25
Finance **	satisfied	5	12	24	27	18
	dissatisfied	13	22	27	12	22

* significant (p<0.1) ** significant (p<0.05)

Table 9

The effect of number of years a practice has utilised computer technology on subtopic rating

Subtopic		1	2	3	4	5
Database/Spreadsheet **	0 years	0	2	5	0	6
	1 year	3	1	9	10	9
	2 years	1	2	6	12	9
	3 years	1	2	4	5	20
	4 years	1	3	5	10	7
	5 years	2	1	7	6	15
	5+years	0	4	16	13	18
Statistics *	0 years	3	1	4	3	2
	1 year	4	6	15	5	2
	2 years	3	8	10	3	6
	3 years	5	4	3	6	15
	4 years	3	5	9	7	2
	5 years	5	3	6	7	10
	5+years	6	12	12	11	11

* significant (p<0.1) ** significant (p<0.05)

CONCLUSION

This case study suggests that for non-computing professionals there is a need to adequately align computing courses to the day-to-day requirements of the professional. It suggests that the involvement of government and academics alone is not sufficient to produce a viable course structure.

The results further suggest that there are a number of factors, related both to the organisation and experience with computing which impinge upon the individual's rating of curricula inclusions.

Given the growing interest in the areas of usability, in the literature, there is a real need to examine these factors in terms of their effect on perceived training needs.

REFERENCES

Ang A.Y. (1992) Information systems Curricula: A Southeast Asian Perspective **Journal of Computer Information Systems**, p. 7 - 15

Ang A.Y. & Lo B.W.N. (1991a) "Changing emphasis in information systems curriculum: An Australian Perspective" **Proceedings of the second conference on Information Systems and Database Special Interest Group**, Australia, 2: pp 339 - 355

Ang A.Y. & Lo B.W.N. (1991b) "Changing emphasis in information systems curriculum: An Australian Industrial Perception" **Australian Computer Conference**

Bergeron F., Raymond L. & Rivard S. (1992) "Organisational benefits of electronic data interchange: An empirical study" **Proceedings of the Third Australian Conference on Information Systems** Wollongong, pp 563 - 578

Dawkins J.S. (1988a) "Higher Education: A Policy Statement" Canberra, Australian Government Printing Service

Dawkins J.S. (1998b) "Industry Training in Australia: The need for change" Canberra, Australian Government Printing Service

El-Khawas E. (1985) "Campuses weld the corporate link" **Educational Record** Spring, pp 37 - 39

Hedberg J. & Harper B. (1992) Information Systems Strategy in the Small Educational Organisation" **Proceedings of the Third Australian Conference on Information Systems** Wollongong, pp 721 - 732

Igbaria M. (1993) User acceptance of microcomputer technology: An empirical test **International Journal of Management Science** 21: pp 73 - 90

Khan H & Robertson I.T. (1992) "Training and experience as predictors of job satisfaction and work motivation when using computers: A correlation study" **Journal of Behaviour and Information Technology** vol. 11, pp 53 - 60

Lo. B.W.N. (1991) "Australian Information Systems Curricula" **The Journal of Computer Information Systems** 31:pp 20 - 33

MacGregor R.C. & Cocks R.S. (1991) A non technical methodology for the purchase of software applicable to the veterinary practice **Aust. Vet. Practitioner**. 22: pp 104 - 108

MacGregor R.C. and Clarke R.J. (1988) 'Loss of the "informal" in Systems Design' in Bullinger H.J., Protonotarios E.N., Bouwhuis D. and Relm F. (eds.) **Proceedings of EURINFO '88** Athens,

Neergaard P. (1992) Microcomputers in small and medium-size companies: benefits achieved and problems encountered **Proceedings of the Third Australian Conference on Information Systems**, Wollongong p 579 - 604

Raymond L (1988) "The impact of computer training on attitudes and usage behaviour of small business managers **Journal of Small Business Management** vol. 26, pp 8 - 13

Seeborg I.S. & Ma C. (1989) MIS Program meets reality: A Survey of Alumni from an Undergraduate Program **Interface** 10: pp 51 - 60

Sharp C. & Lewis N. (1992) "Corporate memory and management information systems" **Proceedings of the Third Australian Conference on Information Systems** Wollongong, pp 149 - 160

Stauffer M. (1992) "Technological change and the older employee: Implication for introduction and training **Journal of Behaviour and Information Technology** vol. 11, pp 46 - 52

Turner J.A. and Karasek R.A. Jr (1984) "Software Ergonomics : effects of computer application design parameters on operator task performance and health" **Ergonomics**, vol. 27, No. 6, pp 663 - 690

Waldhalm S.J. (1991) Client database and written communications **Proceedings of the Eastern States Veterinary Conference**, Orlando, p 637 - 640

Wattenberger J.I & Scaggs (1979) "Curriculum revision and the process of change" in Cohen A.M. (ed.) **Shaping the curriculum** Jossey Bass, San Francisco, pp 1 - 10

Williams T.A. (1992) "Information Technology and Inter organisational Change" **Proceedings of the Third Australian Conference on Information Systems** Wollongong, pp 295 - 308

Yap C.S., Soh C.P.P. & Raman K.S. (1992) Information System Success Factors in Small Business **International Journal of management Science** 20: pp 597 - 609

The Impact of Client/Server Technology on the MIS Curriculum

Robert Schultheis
Management Information Systems
Southern Illinois University
Edwardsville, IL 62026
schulthr@daisy.siue.edu

ABSTRACT

This presentation identifies factors found in practitioner literature that encourage and inhibit development of client/server applications. Skills, knowledge and abilities needed for client/server application development are defined and suggestions for a curriculum to prepare professionals for client/server systems development are described.

INTRODUCTION

Client/server technology has swept the field of management information systems swiftly. Such systems are either being implemented or are being seriously considered by nearly every organization in the U.S. It is not difficult to see why this is so from top management's viewpoint. Client/server systems offer the promise of serious cost reduction, allowing organizations to eliminate or contain mainframe operations and substitute in their stead a distributive computer technology that appears to be much less expensive in hardware, software, and development. In addition, client/server systems offer scalable architecture, making growth easier to manage and less costly. Also, client/server systems, because they are usually GUI based, are substantially more user-friendly to end users than the legacy mainframe systems they replace.

However, the promises of client/server systems may not be fulfilled to the extent that proponents envision. For many organizations, there are substantial costs to install and connect the local area networks that replace mainframe systems, and more costs to manage and maintain these networks. Furthermore, GUI applications often require major and costly upgrades to an organization's personal computer inventory. There are also the costs associated with any new system created by operational disruption. In addition, there are major concerns about using such systems for mission-critical applications, given that the technology is relatively immature, that many personnel lack the experience with client/server application development, and given the additional burdens that distributed systems place on the maintenance of data integrity and system security.

An indirect benefit to management information systems generated by client/server systems has been to return greater control over systems planning, acquisition, development, operation, and management from end users to professionals. Client/server systems are simply too complicated for most end users to digest properly. Thus, in many organizations, management information systems professionals are regaining some control over departmental computing.

One of the costs involved with client/server technology and a major concern of management information systems professionals has

been the retraining of existing systems personnel. Professionals who approach client/server development for the first time are often taken aback by the breadth of skills and knowledge that is required to build these systems. Expanding
the knowledge base and skill levels of developers becomes an urgent requirement.

The reason that management information systems personnel find client/server systems daunting is that they are integrated information systems. Integrated information systems are those that utilize several major technologies to solve business problems. Client/server information systems are considered integrated because they depend at a minimum on PC technologies, LAN technologies, relational database technologies, distributed systems technologies, and application development technologies. They frequently include graphical application development and object-oriented technologies and may also include mainframe and minicomputer technologies. Thus, it is unlikely that any one person will be expertly versed in all of the technologies needed to produce a client/server system. The breadth of technologies is simply too wide. A more appropriate goal is to insure that development teams be comprised of individuals with depth in at least one of the technologies and sufficient understanding of the others that they can work together intelligently.

SUGGESTED TOPICS

The practitioner literature suggests a number of topics that should be considered for the training of client/server developers.

1. **Graphical application tools.** Many client/server systems use graphical-user interfaces to increase the user-friendliness of the end product. Thus, knowledge of and basic skills in the use of graphical application tools, such as Power Builder, are frequently essential to effective application development.

2. **Networking.** Knowledge of networking hardware, software, topologies, management tools, access methods, design, and diagnostics is essential because client/server systems demand networks as their basic platform.

3. **Distributed systems.** It is essential that client/server developers understand the nature of distributed systems and the issues relating to designing and securing them. It is only by distributing the workload across network computing resources that mainframe platforms can be replaced.

4. **Query and object-oriented programming languages.** A common query language used with relational databases common to client/server systems is SQL. Knowledge of SQL and especially how it can be embedded into the code of legacy systems is an essential skill for client/server developers. At the same time, object-oriented programming tools are growing in use, especially as applied to networked applications. Some knowledge and experience with these tools are very useful.

5. **Relational databases.** One of the major elements in any client/server systems is a relational database backend. It is essential that client/server developers have a thorough understanding of relational database structures, including normalization techniques, and relational database modeling and design. It is also essential that developers have experience working with relational backends, such as Oracle. At the same time, the use of object-oriented database modeling techniques appears

to be growing. Some knowledge, skill, and experience in these topics is useful.

6. **Operating systems**. There was a time when developers might succeed by learning one operating system, such as a mainframe operating system. In a client/server environment, such lack of breadth would prove disastrous. Developers should have depth of knowledge in at least one operating system and also understand how to use and modify files in a number of other operating systems and environments, including single-user, multiuser, and local area network systems such as UNIX, MS-DOS, Windows, and Novell Netware.

The chart that follows depicts suggestions for the content and sequencing of topics that would be important for students if they wish to prepare for positions in which client/server systems are developed. A capstone course that would allow students to synthesize their skills and knowledge in the various topics and actually to plan, design, and build simple client/server systems is recommended.

CLIENT/SERVER CURRICULUM GRID

Content Area	Elementary Level	Intermediate Level	Advanced Level
Application Development	Traditional structured systems design methodologies (Gane and Sarson)	CASE methodologies (IEF)	Graphical application methodologies (Power Builder)
Communications, and computer architectures	Standalone single-tasking and multitasking systems (PCs)	Multiuser and networked system (LANs and minis)	Distributed systems (WANS)
Languages	Third-generation languages (COBOL)	Fourth-generation languages (dBase IV)	Query and object-oriented languages (SQL and C++)
Databases	File structures (ISAM)	Database models (relational models)	Object-oriented database models
Operating systems	Use of DOS, Windows, and UNIX	Creation and modification of systems files (bat, ini, and script files)	Use, creation, and modification of network operating system files

A UNIQUE OPPORTUNITY TO PRACTICE WHAT WE PREACH

Dr. Carol Saunders
Florida Atlantic University

This paper was not available at publication time. The author will have copies available at the Conference.

USING ELECTRONIC COMMUNICATIONS MEDIA FOR INTERNATIONAL STUDENT PROJECTS

Joan K. Pierson, Professor
Karen A. Forcht, Professor
Kaushal Chari, Assistant Professor
IDS Department
James Madison University
Harrisonburg, Virginia
and
Robert MacGregor, Senior Lecturer
University of Wollongong
Wollongong, NSW, Australia

ABSTRACT

Management information systems students today are more likely than ever before to take positions with multinational corporations or organizations that do business on a global scale. With an increasing need to internationalize business school networks and the decreasing availability of resources for such endeavors, new educational delivery systems must be found to provide students with international experiences. Electronic communications media provides a cost effective means of equipping students with skills that are important to them as they graduate and take their places in the organizations that deal in the global marketplace. Students at colleges and universities in separate countries can work on joint projects by communicating electronically with one another. Not only do students gain practical experience in working with others via electronic communications, but they gain insights into challenges faced by organizations as they move into the global economy.

INTRODUCTION

Technological developments, increasing international trade, integration of world capital markets, and the emergence of highly competitive production and trade centers in the Far East and Europe have reinforced the realization that the United States must expand its horizons and respond to the challenges of an ever-changing and more complex world economic order [9, 11]. Business schools throughout the world are internationalizing their programs and faculties in response to these major business environmental changes, and, indeed, in countries such as Denmark, France, and Australia, this internationalization is financially supported at federal levels [2, 4, 7]. Challenges faced by schools moving toward internationalization include ensuring that appropriate programs and courses are delivered, recruiting qualified faculty, and attracting qualified students [15].

The discipline of computer information systems is not exempted from the need to internationalize. Throughout the world, computer users approach a computer system with a specific set of cultural requirements [16]. Each nation has developed its own culture, and some areas of the world share a cultural background. Some of the cultural differences such as the presentation of calendar dates and currency symbols are obvious and present little consideration. Others, however, reach much deeper into the history and sensitivity of the country [6]. An example of the difficulty in bridging cultural differences is found in the story of localizing Lotus 1-2-3 into Japanese. The Japanese date counts the year from the ascension of the emperor to the throne, and the 1-2-3 developers included in their test product the ability to reset the counter and to modify the field naming the reign. The Japanese users opposed the feature and requested that it be removed since it anticipated the death of the emperor [8].

Extensive work is being done to develop internationalization standards that will allow developers to create applications that are neutral with respect to language and cultural information [12]. This will ease the task of coding for software developers. Managers, analysts, and designers, however, who work in an international arena need deeper knowledge of cultures to handle their responsibilities effectively.

Of course, it is not the objective of colleges and universities to train students for specific international positions. There is, however, an obligation to provide students with at least some familiarity of and/or experience with the international workplace.

It is unfortunate that at the same time that the educational needs of business students are expanding, budgets either continue to be tight or are being tightened at many colleges and universities. Business schools must explore new educational delivery systems and consider restructuring curricula to successfully meet the pressing needs of students.

Internationalization Approaches

Two approaches to internationalizing a curriculum exist: including international topics in the classroom and providing actual international experiences to students. The former is considerably less costly than the latter. There are many advantages to including both approaches in an information systems curriculum.

Classroom Approach. The classroom approach can be implemented in several ways: a separate international information systems course can be included in the curriculum; international topics can be integrated into the core information systems curriculum; international topics can be incorporated only in select information systems courses [5].

Experiential Approach. There are also several avenues from which to choose that provide university students with worthwhile actual international experiences if cost is not a consideration: semester-abroad and year-abroad programs, international internships, and international tours.

Semester-abroad and year-abroad programs have been popular for years. The nature of study abroad programs vary. Institutions of higher learning in two countries may have exchange agreements or may enter into agreements whereby one institution provides specific courses for students from another. Sometimes a university simply conducts courses in a foreign country. Regardless of the structure of the program, these semester- or year-long approaches have the advantage of permitting students to immerse themselves in a foreign culture for an extended period of time.

International internships during which students gain actual work experience in foreign countries are now in vogue. Again, the student profits from spending an extended period of time in a foreign culture.

Special-interest tours of countries led by faculty members who are familiar with foreign business communities provide short-term international experiences for many students. Tours are generally shorter in duration than the previous options, and students usually do not enjoy the same degree of interaction with the people of the countries which they are visiting.

All of these approaches are expensive. Students usually pay the lion's share of the expenses, however the costs borne by the sponsoring universities cannot be considered negligible. Administration, faculty, overhead, and travel expenses for such programs add up quickly and substantially.

Creative and lower cost approaches are needed in addition to the programs already in place to ensure that opportunities to interact with people in other countries in meaningful ways are provided to all students, not just those who have the financial wherewithal to travel.

ELECTRONIC COMMUNICATIONS MEDIA

Faculty are increasingly attracted to the potential of electronic networks to transform scholarly communications. Until recently, they have focused primarily on research and the exchange of scholarly information [10], however, several joint education programs linking the United States and Southeast Asia in language learning have recently been launched[14]. Faculty experience with international networking can be a springboard from which to launch international network projects involving students from two or more institutions of higher learning in different countries. These projects combine the two approaches to internationalization of information systems curricula--the classroom and experiential approaches.

Telecommunications networks such as Internet that are available to most universities plus other electronic media provide means for establishing "electronic communities," composed of individuals with common interests linked by computers. These media enable collaboration between learners by removing time and distance barriers and providing faculty and students with access to information that is not otherwise easily available [3]. They provide a means to bring students and faculty from different parts of the world together in ways that simulate actual work experience in multinational organizations, whether they be governmental or in the private sector.

Candidate Projects

Deciding upon appropriate projects should not be a stumbling block given the current emphasis on project work as an essential part of most university business curricula. Projects such as case studies, systems analysis and design, and market research are common at the upperclass level and can be "internationalized" by selecting teams composed of students from two or more countries. Students can communicate and work can be accomplished by using technologies that are easily available. Not only do projects of this type offer international education for students, they also benefit students in developing skills using electronic communications networks that are used by governments and multinational businesses.

In the discipline of management information systems, selecting international projects should present little difficulty. Joint undertakings in telecommunications courses provide students with realistic training in dealing with the problems of transporting data across country borders [1]. Analysis and design problems allow students to grapple with the details of implementing international systems. Courses that involve managerial issues can be enhanced by assigning students in two countries to compare and analyze cultural differences and their effects on management decisions. Technology disparities are common stumbling blocks in implementing international information systems. Examining these disparities can be incorporated into projects that fall into many classifications.

Communications Media

Three common technologies that support cooperative international students projects are
- -FAX
- -Internet
- -Videoconferencing

These technologies are based on transmitting electronic signals over communications networks. Each technology has advantages and disadvantages. A combination of all three can overcome individual disadvantages and enable students to become familiar with their uses.

The FAX technology involves sending documents electronically over regular telephone lines. A transmission path (circuit) is completed between the sender and the receiver before the FAX is transmitted thereby enabling the receiver to receive the FAX almost instantaneously.

Internet refers to a collection of nearly 10,000 interconnected computer networks that span the entire globe. Internet has evolved from the ARPANET, a network created by the U.S. Department of Defense Advanced Research Project Agency in 1969. Computers in universities, government organizations, and corporations are now linked to the Internet. In Internet, messages meant for international destinations are routed over regional, national, and international networks. The messages incur store-and-forward delays while traversing through the Internet depending on the network load. Therefore, unlike FAX, messages sent over the Internet often do not arrive instantaneously. FAX communication is, however, more expensive than Internet.

Videoconferencing technology allows people at various locations to send video signals thereby enabling them to interact verbally and visually. The disadvantage of videoconferencing is its high costs. This is due to the large bandwidth (transmission capacity) required to send moving images along with the audio. One unique advantage of videoconferencing over FAX and the Internet is that participants receive not only verbal communications but can also observe body language, sometimes an important factor in successfully completing team projects [13].

The cost of bandwidth is falling due to the advances in technology and the increased competition in the marketplace. This is making newer technologies such as Integrated Services Digital Network (ISDN) more cost effective. ISDN technology allows the integration of voice, data, and images, thereby facilitating the transmission of multi-media information. Internet presently does not have the capability to handle multimedia applications due to the limitations in bandwidth and switching technology. However, when Internet does become an information superhighway that supports multi-media applications, Internet alone would be the most logical choice for cooperative projects. Till such time, FAX, along with the Internet and videoconferencing can be used in supporting cooperative international student projects.

Requirements For Successful Projects

There are several factors that affect the successful implementation of any international undertaking involving students. In each institution involved, there should be
- -a highly supportive administration,
- -a faculty champion,
- -a core of committed international specialists,
- -supportive faculty members,
- -and adequate resources [17].

Administrative support cannot be overemphasized. Such backing is necessary to foster cooperation among faculties in the institutions involved and to overcome problems that arise. Fortunately, administrative support is likely to be forthcoming in view of the increased emphasis on internationalization of business curricula.

The driving force behind successful joint classroom ventures is usually a dedicated and energetic faculty champion to set up the framework for a successful experience that will ensure educational benefits to students. Initial contacts that may lead to joint projects may come from international professional conferences, joint research efforts, longstanding professional relationships, international faculty exchange, formal agreements between universities, and faculty travel. An essential factor is that at least one faculty member in each participating institution is convinced of the importance of setting up the project and is willing to take responsibility for the preliminary planning that may require a considerable amount of time.

The support of other faculty in the department is also important to encourage students to participate in projects that may take far more time and effort than are normally required. The venture is more likely to succeed if academic advisors and other support personnel are involved during the planning and implementation phases of the projects.

Finally, adequate resources must be available. Although the costs associated with joint international projects that use electronic media are far less that of other international student experiences, there will be expenses that may include travel, administrative overhead, and communications.

The projects themselves are also important to the success of cooperative ventures. They must fit into the curricula of both institutions. If these elements are in place, the problems of different academic calendars, time zones, student cultures, and methods of evaluation can be addressed.

Benefits to Students

Finally, what are the benefits to the students? They include

-experience of working with students from other countries,
-insights into other cultures,
-effects of culture on the business community,
-establishing relationships that may last longer than the span of a semester or an academic year,
-gaining an understanding of the benefits of membership in an electronic community, and
-experience with international communications networks.

Most of these benefits can be gained by students through many means. The advantage of the use of electronic communications technology to support international projects is the number of students that can be served at a relatively small cost both to students and their colleges or universities.

CONCLUSION

Tremendous changes have taken place in the way business is conducted in the past thirty years. Corporations have increasingly sought foreign markets for their goods and services and many have grown from primarily national concerns to multinational organizations. Trade agreements are breaking down national trade borders and changing the manner in which business is carried out. To respond to these changes, business school curricula are changing. An important change taking place in many business schools is the "internationalization" of the curricula. It is common for economics, marketing, management, accounting, finance, business law, and information systems courses to include international topics. Some business schools require international travel for their students; others are looking at alternatives to provide international student experiences.

With the expanding availability and falling costs of electronic communications, joint student projects using electronic technology may prove a cost effective alternative to foreign travel with the aim of acquiring international computer information systems experience for many students. Such projects combine the two approaches to internationalizing the information systems curriculum--the classroom approach and the experiential approach.

REFERENCES

[1] Ameritech Report, "Superschools: Information in the Information Age and Beyond," Ameritech Foundation, 1992.

[2] Beeson, G. W., D. M. Stokes, and H. C. Symmonds, "An Innovative Higher Education Course to Meet Industry's Needs," Journal of Higher Education Research and Development, Vol. 11, No. 1, 1992, pp. 21-38.

[3] Connolly, Frank W., and Chuck Schneebeck, "The Community of Electronic Learners," Educom Review, Vol. 28, No. 3, May/June, 1993, pp. 36-37.

[4] Dawkins, J. S., "Higher Education: A Policy Statement," Canberra: Australian Government Printing Service, 1988.

[5] Goslar, Martin D., and P. Candance Deans, "A Pilot Study of Information Systems Curriculum in Foreign Educational Institutions," Journal of Computer Information Systems, Vol. XXXIV, No. 1, Fall, 1993, pp. 8-22.

[6] Greenwood, Timothy, "International Cultural Differences in Software," Digital Technical Journal, Vol. 5, No. 3, Summer, 1993, pp. 8-20.

[7] Hansen, L. A., "A More or Less Happy Relationship: Industry University Cooperation in Denmark," Journal of Engineering Education, Vol. 10, pp. 127-131.

[8] Hapgood, F., "A Journey East--The Making of 1-2-3 Release 2J," Lotus: Computing for Managers and Professionals, Cambridge, Massachusetts: Lotus Development Corporation, 1987.

[9] Light, Timothy, "Erasing Borders: The Changing Role of American Higher Education in the World," European Journal of Education, Vol. 28, No. 3, 1993, pp. 253-272.

[10] Peters, Paul Evan, "Exploring the Potential to Transform Scholarly Communication," Educom Review, Vol. 28, No. 2, March/April, 1993, pp. 18-21.

[11] Philippatos, George C., "Benefits of an International Education: Experiences in Greece and Chile," Survey of Business, Vol. 25, No. 2, Fall, 1989, pp. 36-37.

[12] Rannenberg, Wendy, and Jurgen Bettels, "The X/Open Internationalization Model," Digital Technical Journal, Vol. 5, No. 3, Summer, 1993, pp. 32-42.

[13] Ravichandran, Ramrathnam, and Nazim U. Ahmed, "Offshore Systems Development," Information & Management, Vol. 24, No. 1, January, 1993, pp. 33-40.

[14] Savage, J. G., and D. J. Wedemeyer (eds), "Pacific Telecommunications Council Annual Conference Proceedings," Honolulu, 1993, p. 1102.

[15] Smart, Carolyne, Terry Ursacki, Ilan Vertinsky, and Donald Wehrung, "Administrative Structures and Processes to Promote Internationalization," Journal of Business Administration, Vol. 21, No. 1, 1992/93, pp. 265-280.

[16] Standaert, R., "Technical Rationality in Education Management: A Survey Covering England, France, and Germany," European Journal of Education, Vol. 28, No. 2, 1993, pp. 159-176.

[17] Steers, Richard M., and Gerardo R. Ungsen, "In Search of the Holy Grail: Reflections on the Internationalization of Business Education," Journal of Business Administration, Vol. 21, No. 1, 1992/93, pp. 301-316.

TEACHING STUDENTS TO BE MORE CREATIVE

J. Daniel Couger
Distinguished Professor
Information Systems
University of Colorado, Colorado Springs

Several modules on creativity have been added to the updated curriculum for undergraduate programs in Information Systems, prepared by the joint team of DPMA-ACM-AIS and reviewed in another session in this ISECON conference.

Yet, few faculty have had the opportunity to teach the subject of creativity. In this session I'll be sharing my experience in teaching a full course on the subject for 6 semesters and teaching modules in other courses for 8 semesters. I experimented with a variety of teaching approaches until I came up with an approach that I believe is best suited for information systems majors.

It is a very interesting topic to teach -- one I've enjoyed more than any other I've taught. Almost all people want to improve their personal creativity, so students are internally motivated to learn the subject.

In this session we will discuss what to teach and how best to teach it. We will cover concepts, models, methodologies and techniques of creativity. We will also discuss improving team and well as individual creativity. Another important topic is climate for creativity. Both managers and teammates impact climate -- they can make the climate supportive or repressive for creative expression.

Creativity is considered by some to be unteachable. Yet, the research proves conclusively that creativity can be taught and that significant improvement can result. In my own experience in introducing creativity improvement programs in IS organizations, ROI of 200% to 600% was obtained. There is a very systematic framework for teaching creativity, the CPS (creative problem solving) methodology. We will discuss how to teach the topic within that framework.

A FRAMEWORK FOR STUDENT-CENTERED APPROACH TO LEARNING IN AN INFORMATION SYSTEM CURRICULUM

G. Joyce Dunston
Julio Gallardo
Data Processing Department
Hostos Community College of CUNY
Bronx, NY 10451
jagho@cunyvm.cuny.edu

ABSTRACT

This paper discusses the framework for a student-centered curriculum in information system. The foundation of the curriculum rests on two popular instructional methods: collaborative learning and writing to learn. The framework is an outgrowth of current research findings and the Data Processing Department's ten year effort to meet today's educational challenges.

INTRODUCTION

Across the country, educational institutions are reshaping policies and structures in order to energize their academies (American Association for Higher Education Research Forum, 1994). Likewise, the Data Processing Department of Hostos Community College is restructuring its curriculum so that a student-centered approach to learning is systematically employed throughout. Collaborative learning and writing to learn activities that develop critical thinking skills are primary elements of the curriculum framework. Using research findings regarding the appropriate application of each technique, the Department is creating instructional materials for use both inside and outside the classroom.

COLLEGE ENVIRONMENT

Hostos Community College is located in the South Bronx and was the first unit in the University System deliberately planned for and placed in an economically depressed area. Its primary responsibility is to meet the academic needs of the members of the community. Community residents face severe financial hardships and linguistic barriers, particularly in the case of those who do not speak English. Unemployment is rampant and a large number of residents receive public assistance.

In general, there are two types of students at the College: holders of General Equivalency Diplomas (GED) constitute 30% of the student population, and the rest are traditional high school graduates. Within the two groups there is significant diversification. For example, some GED holders dropped out of high school prior to graduation and are generally in need of remediation. In contrast, others have completed high school in another country, but took the GED because their academic records could not be obtained. Diversity among the group holding traditional high school diplomas is also extensive. Most of the College's student population comes from high schools throughout Latin America and New York City. In both areas the quality of the high schools vary significantly and is reflected in the students' academic performances at the College. In addition, within this group there are students who entered college directly from high school and those who waited several years before enrolling into college. In fact, 42% of Hostos' student population waited six years or more before entering college, with nearly 24% waiting more than 11 years.

CURRICULUM FRAMEWORK

The Data Processing Department was initiated in 1982. During the course of time various teaching methods have been tried. As a result of these experiences, the Department is currently attempting to achieve a more systematic application of student-centered learning strategies. That is, integrate active-learning instructional methodologies into all courses. To this end, a curriculum framework has been developed, and instructional aids and materials to the support the design are being prepared. The goal is to create a curriculum that "...(makes) real demands for hard work, (provides) real opportunities for students to talk a lot, (ensures) real engagement between faculty and students and (offers) real avenues for the out-of-class use of skills"(Kramer, 1990 p.54).

Prior experience has shown that students enrolled in the Data Processing program at Hostos benefit from a group study approach to learning. This approach provides an excellent support system for them. Furthermore, since critical thinking and writing skills are inadequate among the students, writing and collaborative learning activities that foster critical thinking have been incorporated as integral parts of the curriculum. Also an internship experience serves to expose students to the professional world. This is a vital component of the students' learning experience since many do not have prior experience working in a professional setting.

Collaborative learning activities were chosen as a central component of the curriculum for two other reasons. First, is the belief that working on group projects facilitates student adaptation to the teamwork activities encountered in the work place. Second is the view that collaborative learning has the potential to create an environment where an interactive mode of teaching and learning exist. Such an environment, Johnnetta Cole has noted, enables students to be fuller participants in their education (Cole, 1994, p.27), thereby laying the foundation for lifelong learning.

The decision to include writing in the curriculum was reinforced by researchers' findings that students become better writers the more they write and when they write "...as a means of achieving a goal that includes more than simply...writing well"(Parker and Goodkin, 1987, p.33). Therefore, the use of well-structured writing activities in all courses is a focal point in the curriculum's design for the purpose of teaching information system concepts while at the same time reinforcing writing skills.

Normally students who enter the curriculum lack adequate problem solving skills. They generally rely on the brute force approach. That is, they come up with a solution, if it doesn't work something else is tried. This is repeated again and again in an attempt to arrive at a correct solution. In most cases, there is no attempt to understand why an action did not work or, for that matter, why it did. Hence, the curriculum framework includes the use of critical thinking exercises in all courses to help students think about information. In this way, "...students come to realize that a discipline is not simply a repository of accumulated knowledge, but is instead a dynamic, creative thinking activity---a structure of concepts and methodologies used to organize experience, approach problems and give explanations" (Chaffee, 1993, p 24).

Collaborative Learning

Collaborative learning is a very rich activity. Its effects are multiplicative. That is, "People themselves learn when they teach others...They gain an active knowledge of what they had before known only passively, and they become aware of their ignorance in a practical way which is the necessary first step to learning more" (Bruffee, 1973,p 641).

When a collaborative learning approach is employed students have to learn to learn collaboratively. The instructor must set up the structure. Those experienced in the use of the technique suggest "Beginning with specific questions on the material at hand,...then pose broader questions, and eventually propose that groups begin discovering the important problems and questions on their own" (Bruffee, 1973, p 642).

Collaborative learning involves profound changes in the ways in which instructors and students work. As such, it is important:

1. to start slowly and go slowly enough to recognize changes as they take place;
2. to make sure instructions for all assignments are clear and tasks are appropriate; and
3. to ensure that time is allowed for feedback, for example allow time for students to report on their group's work or time to reflect on the group experience.

Finally, the instructor must recognize and accept his/her new role. No longer is he/she the "principal conductor, the catalyst, the guide or synthesizer of class discussion" (Feola and Miller, 1986, p 3), but rather an organizer of people into learning communities "...in which the teacher moves to the perimeter of the action, once the scene is set" (Bruffee,1973,p 637). It is the instructor's role to know what learning is going on in the groups, "...to set appropriate tasks for groups, find appropriate means to accomplish them, and watch groups at work to see how to handle them"(Feola and Miller, 1986, p 3).

For the most part, collaborative learning experiences are found in all the courses that constitute the Data Processing curriculum at Hostos Community College. Collaborative learning activities are most prevalent in the programming and programming related courses and generally relate to program and system development activities. For example, in the beginning programming courses, where students are assigned very small-scale projects, members of the groups share ideas about how the project should be approached. In the end each is expected to arrive at and submit his/her own solution. In addition to helping each other explore the concepts to be learned, this process helps to demonstrate that a given problem can have more than one solution. In the intermediate courses, i.e. COBOL and spreadsheet design, the projects are more extensive than those in the beginning courses. In these courses students have two types of group experiences. In one case all groups work independently to solve the same problem. In other instances groups work on separate components of a problem which are then combined to produce a final product. The latter mimics, more closely, work in the professional world as the final product depends on each teams productive efforts. However, both experiences allow students to share ideas and evaluate competing solutions. In advanced courses students are given more extensive team projects to complete. In these instances groups are assigned the same type of projects i.e. analyze business needs. The students are responsible for managing and directing the work, determining such things as who is to play what role, the activities that are to be performed and the schedule of activities. The instructor oversees the work establishing guidelines and milestones.

Writing to Learn

Articulating and learning are inseparable activities. (Knoblauch & Brannon, 1983), thus writing activities facilitate student's learning. However, various types of writing assignments/tasks yield differing results. For instance, Hayes and Simpson note that the kind of writing affects the kind and quality of learning. That limited writing, i.e. note taking, answering short-answer review questions, leads to limited engagement with the material. While analytic writing "...encourages more thoughtful attention to a smaller amount of information (where) students...consider fewer ideas, but...understand and link ideas in more cognitively complex ways"(Hayes and Simpson, 1994, p 15). Thus, writing to learn proponents promote the development and use of writing assignments that make writing a vehicle for exploration and discovery, rather than as a means of testing students' knowledge of a subject.

According to D.L. Pearce of Eastern Montana University, "Writing on a topic being covered in class requires students to examine the concepts and facts involved, to focus on and internalize important concepts, and to make these concepts at least to some degree their own"(Tchudi, 1986, p 20).

The results of writing to learn activities do not require extensive instructor comments or response, since what is written is primarily for the benefit of the writer as an aid to clarifying the subject matter. Writing to learn activities moves the focus of writing evaluation from style and grammar to content: what is being said, not how. Therefore, "In only one sense should the content instructor attend to writing quality: vague, murky, incoherent, jumbled prose maybe an indication that the student is missing course content. The writing quality is thus a reflection of content mastery" (Tchudi, 1986, p. 23).

Although significant progress has been made, much remains to be accomplished to fully integrate writing into the Data Processing curriculum at Hostos. The Departments goal is to include writing activities in all the courses so as to facilitate learning and improve students writing abilities. Since different types of writing experiences make possible different kinds of thinking (Parker and Goodkin, 1987), the first task is to establish writing objectives for each course. Thus far, this has been completed for only two of the courses: the introductory and exit courses.

Once objectives have been established, the next step is to develop an appropriate set of activities for each course. In doing so, instructors concern about the time required to perform, evaluate and review the exercises must be addressed. Often instructors fear that the inclusion of writing in the curriculum will negatively impact their primary responsibility---teaching the discipline. These fears, however, are largely unfounded as evidence shows that students who use writing in their everyday school activities learn material more effectively than those who do not, and that there exists ways to promote writing in the classroom that do not require large amounts of class time and extensive theme correction by the instructors (Smit, 1991, p. 2). Therefore it becomes critical to the success of the endeavor to develop writing exercises that help students explore the material and require them to use the concepts taught in the course.

The writing objective established for the introductory course is to reinforce writing skills and course content so that students are able to write brief summaries on various computer topics. Thus the introductory course employs the use of informal writing exercises since informal use of language has been found to be particularly important for the early stages of understanding information and new ideas (Parker and Goodkin, 1987, p. 44). Often the course makes use of focused freewritings. These are exercises where students write free associations to whatever comes into their minds in response to readings, lecture or discussion (Smit, 1991, p. 2). For example, at the beginning of the class students are assigned to write for five minutes on a topic found in the reading given for homework or a topic covered in a prior session. Other types of writing exercises used in this course can be categorized as microthemes. Microthemes are defined as

...mini-essays which require students to write summaries...pose questions, work with data... (Smit, 1991, p. 2). Most often these are done in group. For example, groups may be asked to write a letter to the author stating what they liked or disliked about the way in which a topic was covered. The letter is then used as a basis for the lecture. Or groups may be asked to write the author any questions they would like to ask about a topic. The groups exchange questions and attempt to answer another groups set of questions. Like the letter, the questions become the basis of the lecture.

For the exit course, Internship, formal use of language is required. According to Parker and Goodkin when ...information, concepts, etc....are more fully grasped, it becomes appropriate then to communicate (ones) understanding in more public, conventional forms of discourse (Parker and Goodkin, 1987, p.44). For this reason students enrolled in the internship course are required to write reports and make presentations on current information system topics.

Critical Thinking

To date, no systematic, curriculum-wide approach, to the integration of critical thinking has been attempted. Each instructor uses his/her own means to incorporate the development of critical thinking skills in a course. Generally this entails:

- starting with simple problems that clearly identify the facts and objectives, then progressing to more extensive problems where the information has to be discerned
- promoting work in groups and group discussions where students are strongly encouraged to focus first on understanding the problem before developing a solution
- repeating the same information system concepts and types of problems, adjusted to the level and nature of the course, with the intent that students learn to use and build on prior experiences.

CONCLUSION

Key to the development of the curriculum framework was the creation of learning objectives for the program, which in turn, were used as guideposts for the establishment of learning objectives for each of the courses. In this way, the courses were designed to build on each other with regard to the set of concepts, pertinent to the profession, to be taught; the development of writing and critical thinking skills; and the use of collaborative learning. While there has been significant effort to develop and integrate collaborative learning activities in the curriculum, more attention needs to be directed toward the inclusion of critical thinking and writing in the curriculum. More theoretical research in these areas would also be beneficial.

Although the use of writing and collaborative learning activities reduces the time that would otherwise be allocated to the coverage of content, their incorporation into the classroom is nevertheless deemed beneficial. In the words of Richard Paul, ...students are more likely to profitably extend their learning ... if they learn a few basic principles and concepts deeply through multiple, reasoned exposures, than if they are given so many concepts that, in the end, they grasp none deeply or well (Paul, 1993, p. 11).

REFERENCES

American Association for Higher Education Research Forum (1994). A Research Agenda for Envisioning the 21st Century Academic Workplace Through Responsive Academic Citizenship.

Browne, Neil & Keeley, Stuart (1988). Do College Students Know How to Think Critically When They Graduate? Bulletin, 3(1). 1-2.

Bruffee, Kenneth (1973). Collaborative Learning: Some Practical Models. College English. 634-643.

Chaffee, John (1993). Teaching for Critical Thinking. Change. 24-25.

Data Processing Unit Curriculum Committee (1993). Microcomputers for Business. Hostos Community College, Bronx, New York.

Feola, Maryann and Miller, Peter (1986). Practicing Collaborative Learning. College of Staten Island (CUNY).

Hayes, Christopher and Simpson Michele (1994). The Effects of Extended Writing on Students' Understanding of Content-Area Concepts. Research in the Teaching of English. 13-29.

Hostos Institutional Review Committee (1992). Academic Program Review. Hostos Community College, Bronx, New York.

Hunter, Beverly (1986). What is Fundamental in an Information Age? A Focus on Curriculum. Topics in Computer Education: National Educational Computer Policy Alternatives. 118-128.

Kramer, Martin (1990). Curriculum Reform Rules of Engagement. Change. 54.

Parker, Robert & Goodkin, Vera (1987). Language and Thinking. The Consequences of Writing: Enhancing Learning in the Disciplines. Upper Montclair, NJ: Boyton/Cook. 29-48.

Paul, Richard (1993). Overcoming the Addiction to Coverage. Change. 11.

Smit, David (1991). Improving Student Writing. Idea Paper No. 25. Center for Faculty Evaluation and Development.

Tchudi, Steven (1986) .Teaching Writing in the Content Areas: College Level. NEA.

Teaching Geographic Information Systems

James B. Pick*, Roger Baty*, and Michael Phoenix**

*University of Redlands, 1200 East Colton Avenue
Redlands, California 92373-0999

**Environmental Systems Research Institute
380 New York Street, Redlands, California 92373-8100

ABSTRACT

Geographic Information Systems (GIS) is an important and rapidly growing area in business and government that falls in between the academic fields of geography, environmental science, and information systems. This paper examines GIS in the context of the body of knowledge, courses, and curricula in information systems. It illustrates the potential role of GIS by a case study of developing and implementing a 3-unit elective course in GIS in the information systems program at University of Redlands. The case examines the course development process, corporate involvements, GIS student design project, course content, and results and outcomes. The paper recommends ideas for teaching improvement in GIS courses and suggests how GIS may evolve further as a new component in information systems and management programs.

INTRODUCTION

GIS has emerged over the past decade as a new and rapidly growing business area. A GIS is an information system that has spatial components and is utilized for modeling and decision support and is supported by a data-base incorporating geographic data structures. These systems, originally centered in national, state, and local governments, have blossomed into widespread use in diverse areas of business including marketing, transportation, distribution and others.

This paper focuses on the teaching of GIS in the information systems (IS) and management disciplines. This course is a new one for the academic management setting; it is currently approved as a course in very few business school curricula. After placing GIS in a broad context across academic disciplines and curricula, the paper presents a case study of offering a GIS elective course to information systems and management students at the University of Redlands. This course was developed and taught over a one year period through a partnership of campus-based and corporate faculty. The course included close cooperation between students and faculty and local government and corporations. The paper analyzes the results and outcomes of this course. It then proceeds to discuss teaching introductory GIS in general. It recommends ways to improve the teaching of GIS. It suggests how GIS may evolve further in IS/management.

The objective of this paper is, first, to analyze the place of GIS in the teaching discipline of information systems. In doing this, it is necessary to compare how GIS fits into neighboring disciplines. A second objective is to present a case example of developing a full GIS course for information systems and management students at University of Redllands. Since GIS is barely getting started in IS/management courses and curricula, this case study points out some of the teaching opportunities, challenges, and design and implementation issues in the IS/management context. It conveys experiences makes suggestions in teaching such a course. Among many points, it emphasizes that spatial analysis is a necessary component of a GIS course. This case study may be useful to academic programs and faculty anticipating initiating or improving GIS in courses offerings and curricula. The next goal of the paper is to make recommendations on the teaching GIS in IS/management programs. Finally, the paper proposes several ways that GIS might evolve further in the future, as part of IS/management programs and curricula. This paper section may be useful to IS administrators and faculty who appreciate the importance of this new field and are planning to expand coverage of the field in IS and management programs. Since GIS is a new and emerging field, these recommendation are intended to stimulate ideas, rather than to represent a formalized approach.

GEOGRAPHIC INFORMATION SYSTEMS IN ACADEMIC CURRICULA

GIS is taught in several undergraduate areas within universities. Its location implies fundamental differences in course objectives and course content, as well as the in educational background and preparation of students taking the course. GIS was originally taught in geography departments. This is in part because the leadership in the early field of GIS was dominated by geographers.

In the field of computer science, GIS is rarely taught as an independent course, and is not included in the 1991 Computer Science national model curriculum (Tucker et al., 1991. Rather than being formalized as a course, the place of GIS in Computer Science is as an emerging technology application that can supplement teaching and curricula.

For instance, GIS has been the focus of a one week lecture/laboratory segment of the introductory computer science course at Louisiana State University (Hall et al., 1993). In this instance, students learned general concepts of GIS, saw GIS in action through videos, and utilized a very simple and inexpensive mapping package (PC Globe) in several lab exercises. The package was used to produce maps and several two-coverage maps of elevation, climate, cities, and population used pre-entered data. Students were also given a 1-2 hour out-of-class exercise with PC Globe to map a country. This had a further advantage of introducing students to other cultures.

GIS is unlikely to encompass enough of the computer science body of knowledge to warrant a full course. In fact, in computer science curricula, GIS is viewed as one of numerous modern software applications. In elementary courses, it serves as an application example to illustrate and give practice on theoretical concepts in computer science. For more advanced courses in computer science, special GIS topics may be included such as data storage of spatial data, 3-D algorithms, raster/vector conversion, and representation of temporal data.

GIS has a logical place in information systems courses and curricula, since a GIS represents a type of information system. However, IS curricula standards were first established in the 70s and early 80s, i.e. prior to the widespread use of GIS. Hence, GIS was not incorporated into early national model IS curricula, a pattern has continued with the latest national model IS curriculum, the DPMA IS '90 Curriculum (DPMA, 1991). Although IS '90 is based on a detailed and comprehensive IS body of knowledge, there is no mention of GIS, with the closest categories in the body of knowledge being 2.9.7 Manufacturers, OEMs, Systems Integrators, Software Developers and 4.8.4 Decision Support Systems. Likewise, in two earlier and prominent IS model curricula, the ACM '83 and DPMA '86, GIS was not included as a required or elective course (ACM, 1983; DPMA, 1986).

As in computer science, so far, in IS/management courses and curricula, the role of GIS has been mainly as a lecture topic or laboratory example of a current application area. GIS is being taught in a few IS programs on an exploratory or innovation basis as a full-fledged course. These initial course offerings require extra effort on the part of instructors and students, since there is little precedent in the IS/management context. Many materials including texts, cases, labs, etc. have to be developed. Instructors must engage in learning themselves, in order to develop an appropriate offering.

The next section presents an example of developing a GIS course as an information elective offering to IS and management students.

A CASE STUDY: GIS AS AN ELECTIVE COURSE IN WHITEHEAD COLLEGE AT UNIVERSITY OF REDLANDS

Background on University of Redlands and Whitehead College

The University of Redlands is a private, comprehensive university of about 4,000 students. It has both traditional programs in a full range of liberal arts disciplines for 4 year undergraduate and non-traditional programs in management and information systems for working adults. There are about 2,200 student enrolled in the latter programs in Whitehead College, which include B.S. in Information Systems, B.S. in Business and Management, and MBA. These programs are taught by full-time academic faculty, as well as corporate adjunct faculty. The IS Program has about 200 students; has developed a new curriculum in 1993 (Pick and Schenk, 1993); and stresses involvement of corporations (Schenk and Pick, 1994). In Whitehead College programs, students go through an entire program with the same student group, in a fixed sequence of required courses. Students may take a limited number of elective courses either concurrently or after completing the required course sequence.

The university established a focus in geographic information systems two years ago, a direction furthered by a partnership with a large GIS company. The university faculty with interest in GIS encompass environmental studies, biology, chemistry, economics, and management. The campus administration recognized this interest, and established linkages with Environmental Systems Research Institute, one of the world's largest GIS software firms. Its products include ArcInfo, ArcView, and ArcCAD. The partnership led to ESRI donating a GIS laboratory to the campus, ESRI participation in joint research and instructional projects on the campus, and ESRI's provision of training to over twenty full time faculty and several graduate students.

The Information Systems program emphasizes systems development and IS practice. It has a 54 unit curriculum plus 6 units of pre-requisites. Students enter the program with approximately two years of college credits from other universities or colleges plus at least five years of working experience. The curriculum includes ten IS courses closely patterned after the DPMA 90 national curriculum (DPMA, 1990) and six management and business courses including accounting, finance, etc. (Pick and Schenk, 1993). A major feature of the program is the applied software development project, a nine-month project on a business problem in industry or government, often at the student's work organization. Students and faculty alike tend to regard the project as a cornerstone of the program, a feature that provides students with real work IS problem solving skills under the tutelage of faculty and work-professional experts. There were only two information systems electives, prior to the present case study, in C Programming and Artificial Intelligence.

The university-ESRI partnership constitutes an important factor that has encouraged faculty to develop innovative GIS courses. In Environmental Studies in the traditional college, several GIS courses have been developed and tested, in particular courses on spatial information systems and environmental design studio. The courses have been primarily designed and taught by ESRI professional staff.

Corporate Involvement in Developing the GIS Course

Two of the authors who are full time faculty at the university (Baty, Pick) conceived of the idea for the GIS elective course in Whitehead College and immediately approached ESRI, due to the campuswide partnership and their knowledge of the company from taking the firm's software training workshops. The third author, Michael Phoenix, ESRI's Director of Higher Education Programs and a former geography professor at several universities, was interested in developing a GIS elective course.. The team approach offered a mixture of faculty skills for course development encompassed anthropology, environmental studies, and applied GIS (Baty); information systems, environmental studies, and applied GIS (Pick); and geography, GIS (Phoenix). The course was developed over a period of nine months and subsequently taught by this faculty team.

The corporate teaching partnership provided many advantages in course development. First, it assured that course content would be very up-to-date. This is essential in a field of rapidly moving technology. Many recently published materials on GIS are out-of-date, due to lags in publication and the pace of change. Another advantage to teaming with ESRI was access to expertise on GIS software training. GIS software tends to be more complicated and difficult, and not well suited for beginning instruction. The greater access to ESRI software materials, video and other training materials, and expertise saved many efforts that might have been misdirected. The corporate co-instructor had wide knowledge of GIS instructional efforts in a variety of programs and curricula around the country, which helped place our course development efforts in broader context. Finally, the ESRI faculty's technical expertise on GIS was much deeper than would ordinarily be found in the faculty of a small, comprehensive university.

Since all three faculty have strong interests in environmental studies and applications, we decided to focus on environmental applications of GIS. Such focusing is recommended since GIS applications range widely from government planning and operations to utilities, marketing, transportation, real estate, environment, etc. Given this applications spectrum, it is important for GIS instructors to reduce the application breadth, in order to conserve on classroom and study time. We narrowed the environmental focus even more, to focus principally on water resources and the related area of sanitation. These are critical problem areas for the water-poor southern California metropolis.

Corporate involvement in course development also encompassed other community organizations and expertise. The instructors contacted the San Bernardino Valley Water District, which relies heavily on GIS analysis and modeling (through ESRI contacts). In discussions with the Water District, it agreed to host a field trip for students to observe the district's very successful GIS operation. At the same time, this field trip coincided with the environmental sub-theme of the course, since GIS at the district is primarily utilized to model groundwater capacity and flows. This idea continued to evolve. As the course was finally formulated, a five hour field trip was included, consisting of an orientation talk on the district's water system (given by the chief water engineer), an orientation and tour of the district's GIS facility (given by the GIS specialist), and a two hour bus tour of the district's field sites. Since students were unfamiliar with the district operations, the site tour crystallized the reality modeled by the GIS. Students were given GIS-produced maps, showing several layers of the water system, to take along on the field trip and identify real-world objects and areas on the maps. The field trip was a crucial element in advancing students' GIS knowledge. In particular, it gave students practical background on southern California's water supply; introduced them to the typical equipment, software and people factors in a small GIS department; and demonstrated how GIS can be developed, operated, and managed in a small organization.

Another corporate link for the course was a guest talk on GIS applied to an urban sanitation system. The speaker was the project manager of team that developed a highly successful GIS model for the Los Angeles County Sanitation District. The course instructors met with this manager at his office for half a day to learn much more about the details of his GIS system and to discuss the plan for his class presentation. This provided a learning opportunity for the faculty and a chance for the cooperate speaker to refine the focus and intent of his talk. The talk was given the last session of the course as a capstone. It described the sanitation GIS model, but went much further in stressing the importance of user and management involvement in design, advocating a longitudinal approach, discussing visualization of GIS in presentations, etc. The presentation style was multimedia including video, slides, and charts. This talk provided students with many real-world knowledge about succeeding with a larger GIS project, not only technically but also organizationally.

The Process of Course Development

Designing a new and non-standard course in a rapidly evolving technological field demands special efforts and considerations. First, the instructors budgeted a lengthy, nine month development period, in order to give maximal time for testing, experimenting, comparing, and re-testing. They met on a bi-weekly basis throughout this period to discuss ideas, test GIS and learning concepts, and refine course models and materials. As a consequence, the draft course syllabus underwent five major versions, as the instructors worked and re-worked course ideas.

The major evolutions in course development were as follows:

1. The laboratory software evolved from more demanding exercises in ESRI's full-fledged GIS package, ArcInfo, to less demanding labs in ESRI's GIS "viewer," ArcView. This software package enables an end user to view and manipulate GIS data and map layers (i.e. coverages), but not to design and develop the application. It has a modern, interactive and menu-driven user interface and is fairly easy to learn. This change was based on an expected mix of students from information systems, undergraduate and graduate management programs. Although ArcInfo might be appropriate for advanced IS students, it seemed too difficult for a mixture of management and IS students. This change was strongly confirmed by course results.
2. The application focus of the course was narrowed to environmental issues and even further to water resource issues. We talked about the need for students to have some common understanding of the application area. Narrowing the focus made it possible to include some background readings and talks about the area of application. We realized this would be impossible with all GIS areas, nearly

impossible for environmental problems, but more realistic for water problems.

3. We decided to based the lecture content from a different source than the text. We recognized that this is an unusual step for undergraduates (although more common in graduate courses). The reason for this is the emerging status of GIS instructional materials. We felt that the NCGIA Model Curriculum (Goodchild and Kemp, 1990) provided the most comprehensive and up-to-date body of knowledge in GIS. For lecture, we decided to extract relevant parts of the NCGIA curriculum and present them, with each instructor lecturing in his areas of greatest expertise. The NCGIA only provides notes to instructors, but does not provide materials for students. Since no existing text is based on this curriculum, we adopted a very practical text with a focus on government planning (Huxhold, 1991). It is likely that future texts will be written that conform more to the NCGIA; in that case, lecture and text reading could be joined back together.

4. In the lecture part of the course, we distinguished between spatial analysis and GIS and emphasized both as course components. Spatial analysis refers to concepts and principles that are used to analyze spatial locations and relationships. Examples of spatial analysis concepts are concepts of location, point/line/polygon data, and linkages of nongraphics attributes to graphics data. This is contrasted with the GIS course component covering computer-based set of tools used to perform spatial analysis. We ordered the course to cover the spatial analysis first, followed by computer-based GIS. We felt that spatial analysis is essential to broaden students conceptual understanding and enable them to <u>understand the meaning</u> of computer-based GIS.

5. We refined our concept of the course project. Initially we thought computer-based projects would be appropriate, perhaps making use of different GIS software packages, in particular ArcInfo, ArcView, IDRISI and/or Atlas Pro. However, we increasingly moved away from software projects towards the idea of s team-based <u>GIS design project</u>. In the design project, the student would become a member of a design team of 3-4 students. The team would define a real world water- or sanitation-related problem that could be addressed by GIS. The team would then plan the base map, GIS map layers, types of attribute data, and potential features and applications. However, the team would not be expected to construct the GIS using software. Rather, the team would go through analysis and some design and provide an oral and written report based on design.

Teaching the Course: Instructional Results and Student Evaluation and Suggestions

The course was inaugurated in fall of 1993. This section describes the course as taught, as well as the instructional outcomes and student responses.

The course was opened as an elective to any information systems or management undergraduate and MBA students, as well as to MBA students and alumni. The course had a mixture of undergraduates from both majors and two MBA alumni. Since Whitehead College students' required courses are with homogenous groups of students in the major, such a mixture of student levels and majors was unusual. Students received a detailed questionnaire several weeks prior to the course. It asked them about past experience with GIS, company/job title, use of GIS in the company, computer educational background and experience, geography background, knowledge of California water problems, and interest level in hands-on computer labs as a part of the course. This learning assessment tool proved extremely useful in instruction. For instance, we discovered a high student interest in the lab part of the course, identified some students who had GIS in their companies or planned to use GIS at work in the future. We were able to diagnose a bi-modal range of computing skills backgrounds, with about half of the class proficient in computers and programming, while the other half had modest end-user or no computing skills. Thirty percent of students' firms had GIS, and two students had used a GIS, although simple software packages.

This learning assessment tool confirmed to us that our lab level (elementary and intermediate ArcView) was appropriate. We further sensed from the surveys a moderate to strong motivation to study GIS, with about half the class citing potential job benefits.

Some of the key course features were as follows:

1. The lecture segment covered NCGIA Units 1-4 and 6-9, about a quarter of the NCGIA core curriculum. These lecture segments were rotated among the three instructors.
2. The lab segment included three ArcView I labs. These labs were constructed in a "cookbook" format in which students learned about half of the features

and capabilities of ArcView. Students accessed three GIS data-bases provided by ESRI to the campus, ArcUSA (a socio-demographic data-set on U.S. states and counties), Maplewood (a teaching data-set on a small local community), and Redlands (a city planning data-set for the city of Redlands). The latter data-base was directly meaningful to the students -- a fortuitous result of collaborating with a software vendor located in the campus city! Students used a Sun-workstation laboratory in the campus computer center donated by ESRI.

A drawback to campus GIS lab use at the time was lack of computer center support personnel for GIS. Such lab support is strongly recommended in case studies on GIS teaching facilities as a critical success factor for GIS instruction (Palladino and Kemp, 1991). We thought that containing the labs within class time through "cookbook" format would get around the lack of computer center support personnel. However, it turned out that some students' strong interest and abilities in information systems led them into the lab out of classroom hours for lab follow-ups and project work. This unanticipated strong interest of some students in the hands-on, combined with lack of support personnel, led to several problems in utilizing ArcView.

3. The student project went through the entire length of the course and constituted a backbone for student learning. In the first all-day workshop, students formed three teams, coalescing around pre-selected team leaders. The course had time built in for student team meetings, critique sessions by the class as a whole, set-up sessions for the poster session presentation, and time for the session at the final workshop. Students were expected to communicate and meet on the project outside of class time. With working adult students located all over the Los Angeles basin, this implied special efforts to meet in person and/or confer by telephone. Each team managed to establish a different communication pattern; however, there was a learning curve to this -- a cost involved in using teaming with adult students.

4. Other instruction components included text homework assignments, discussion of text readings, in-class videos, and a guest speaker. The textbook was assigned for reading and evaluated through take-home written questions and in-class discussion. These components were intended to suit the needs of adult working students, who often prefer to take greater responsibility themselves over rote, textbook learning and who generally prefer variety in classroom activities.

The course was taught with ten students in fall of 1993. Three student teams were formed, each focusing on a water or sanitation-related environmental issue. Although largely students progressed smoothly and appeared engaged and stimulated, a problem emerged with one project team of three students who selected the environmental problem of GIS applied to groundwater modeling by same water district visited in the field trip. The group wavered on approach. Although the instructors recommended (but did not require) a strictly design approach, i.e. without hands-on computing, this group requested and were granted permission to do a hands-on project in ArcView based on data to be provided by the water district. The instructors felt confident in the group's technical skills, since two student team members were employed professionally as programmers. The results were frustrating. The team was given the wrong data, which had to be corrected. There were problems at the computer center in loading the data. Once loaded, the coverages were different that expected. There were printing problems. There was only time left before the due date to analyze several attributes, rather than the more complete analysis planned by the team. In spite of these problems, the team did turn in a quite acceptable written report and ably demonstrated its limited hands-on results to the class.

The other project teams followed the prescribed design format and were highly successful. In retrospect one team's frustrating result stemmed from several causes: (1) overconfidence that business programming expertise would carry over to GIS programming expertise, (2) lack of a computer center user support person in GIS, (3) lack of bounds and constraints on the project, i.e. the instructors did not strictly limit the project to manual design or otherwise contain the project scope, and (4) problems in data gathering for GIS, which are typical of the real world, but might be counter-productive for student projects. In particular, GIS projects in industry typically are data-intensive in time, cost, and problems (Huxhold, 1991).

Given the novel and innovative nature of the course, the instructors supplemented the ordinary university prescribed student evaluation by with a two page questionnaire asking for more detailed written appraisal of each workshop and course feature. 100 percent of students responded to the course evaluation and 60 percent to the detailed questionnaire.

The general results were quite positive, with 100 percent of students replying affirmatively that they "would you like to have this instructor again." Eighty six percent of evaluation responses on "how well did the professor do" in particular areas were rated either "superb" or "very good." Likewise the detailed questionnaires were generally very positive, with such remarks as "fine comprehensive course," "the course in general was excellent," and "this class has opened a new door for me and I would like to pursue GIS in the future. ... Thank you for the class."

One area was singled out for teaching and course improvement. Given the one team's project outcome, several students suggested changes in the team project including clarification of project criteria and goal, using a "case study" approach to the project based on a fictional situation and data, and providing more time for the project. One student pointed out, "students need to understand the project as it does not necessarily relate well to previous projects in the program." For some students, creative brainstorming as an aspect of a team project may be unfamiliar, but it can be argued that this experience is beneficial. The instructors' challenge is to convey and justify the approach, and engender student commitment to it, ahead of time. It is interesting that no student suggested dropping the project, but rather making changes in its format.

Another measure of the success of the course is subsequent student career or educational benefits from the course. Four months later, forty percent of the students had a definite benefit, i.e. applying for GIS jobs, doing a senior project on GIS software development in campus admissions using ArcInfo, using GIS to start up a management consulting firm to export American products and services, and applying GIS knowledge and experience to evaluating a GIS system for a city government.

DISCUSSION

GIS should be considered as an important and exciting new area for information systems and management programs. GIS is growing rapidly in the corporate world with mushrooming application areas. It is a form of information system and falls within the realm of information systems, even though it has not yet been formally recognized as a part of the IS body of knowledge in model curricula.

The appropriate role of GIS is either as an elective course in information systems or management programs or as an application segment in other courses. The course is not central enough to the foundation topics of information systems to constitute as a required course in information systems or management programs, unless they are joint programs with environmental studies or geography. As an elective course, GIS can be s standalone course, without pre-requisites as in the case study, or it might require a data-base course as prerequisite, in which case it could have deeper coverage of GIS data structures. Another approach would be to require a geography or environmental science pre-requisite, which could advance the level of coverage of GIS applications.

If GIS is reduced down and taught as an application area in another course, the other course might be data-base (GIS data-base design would be highlighted), decision support systems (GIS is often classified as a form of DSS), or computer simulation (GIS's modeling capabilities would be emphasized). It might also be used for a week sequence in a lengthy introductory IS course or course sequence, i.e. in a format similar to that described earlier for introductory computer science (Hall et al., 1993). In such a mode, GIS could illustrate the foundation concepts of data-base, data management, computer graphics, and/or DSS.

In management school settings, GIS might be included as a part of courses in other functional areas, such as marketing, transportation, or environment and business. In such a format, GIS would be used as a tool to illustrate course concepts or to exemplify functional area applications. For instance, it might be used in a marketing course in lecture or lab to illustrate product marketing areas.

The future evolution of GIS in academic information systems depends on the scale and importance that GIS assumes in business and government. If its rapid expansion continues and it eventually grows to become one of the most fundamental applications of information systems, then academically it should become a commonplace elective, or even required, course and a standard curricular component. However, if its role in the business world is smaller, then GIS may settle into place as an elective course in IS/management programs that emphasize environment, graphics, DSS, and/or linkages with geography and planning programs.

CONCLUSIONS

Geographical Information Systems is an emerging area in business and information systems academic programs. There are few course and curricular precedents. GIS has a lot of potential appeal to students, since it has exciting visual and modeling capabilities. It is also a generic tool that can be applied in a variety of business/government applications including environment, planning, marketing, etc.

Teaching GIS presents many special challenges. This includes identifying a standard body of knowledge in a young area, finding appropriate course materials, decided on extent and level of hands-on laboratory experience, deciding on the inclusion and scope of student projects, assessing time allotments, and interacting with real-world practitioners.

The following recommendations are made to instructors who plan to teach GIS:

1. Allot extra time for course development. GIS is a new area, where you may have to struggle with some ideas and concepts yourself as well as spend more time in course design.
2. Emphasize spatial analysis as part of the GIS course content. Teach the generic concepts of spatial analysis as a separate course component from GIS software and technology.
3. Consider bridging to the corporate/government practitioner community. This may involve course design discussions, co-teaching, field visits, guest lecturing and discussions, vendor/software support, and others. Much of the expertise in GIS is out in the real world, so you can benefit by tapping into it.
4. On the laboratory or hands-on projects; assure that the necessary data can be made easily accessible, and that there are GIS lab support personnel appropriate to the difficulty of the labs. Attempt on the lab side only what is instructionally supportable on your campus.
5. Consider the advantages of including a GIS student project with creative design components. In structuring the project assignment, provide goals, define the scope, and delimit the project carefully.

GIS is a rapidly growing area in the real world that academia has the opportunity creatively to respond to. In its current emergent stage in information systems and management programs in universities, GIS will benefit by innovation and experimentation in teaching and curricular design.

REFERENCES

A.B. Tucker et al. 1991. Computing Curricula 1991, A Report of the ACM/IEE-CS Joint Curriculum Task Force. IEEE Computer Society Press.

ACM. 1983. ACM Recommendations for Information Systems, Volume II. New York, New York: Assocation for Computing Machinery, ACM Committee on Computer Curricula of ACM Education Board.

DPMA. 1986. DPMA Model Curriculum 1986. Chicago, Illinois: Data Processing Management Association.

DPMA. 1991. Information Systems: The DPMA Model Curriculum for a Four Year Undergraduate Degree for the 1990s. Park Ridge, Illinois: Data Processing Management Assoociation.

Goodchild, M.F. and K.K. Kemp. 1990 Introduction to GIS: NCGIA Core Curriculum. 3 Vols. University of California, Santa Barbara: National Center for Geographic Analysis.

Hall, Carol, Krishna Agarwal and Alfred McKinney. 1993. Incorporating a GIS Component in an Introductory Computer Science Course. In, isecon '93 Proceedings: Information Systems Education, The Future of Information Systems, Park Ridge, Illinois, Data Processing Management Association Foundation, 134-139.

Huxhold, William E. 1991. An Introduction to Urban Geographic Information Systems. New York: Oxford University Press.

Pick, James B. and K.D. Schenk. 1993. Development of An Information Systems Curriculum for Non-traditional Students. In, Isecon '93 Proceedings: Information Systems Education, Park Ridge, Illinois, DPMA Foundation, 148-155.

Schenk, K.D. and James B. Pick 1994. Successful Strategies for Corporate Involvement in Information Systems Programs. Proceedings of Inf. Systems Educators Conference. DPMA/EDSIG, 116-121.

Information Systems in the Year 2001

Charles M. Lutz, Ed.D., C.C.P.
Dennis LaBonty, Ph.D.
Business Information Systems & Education
Utah State University
Logan Utah 84322-3515
cml@b202.usu.edu or denlab@b202.usu.edu

ABSTRACT

A survey was conducted of 136 IS management personnel to determine the skills needed by our graduates for success in the work environment of the future. A description of the anticipated work environment and a rank ordered listing of computer skills required for entry-level managers was developed. Finally, the impact on the information systems curriculum was assessed.

INTRODUCTION

A new millennium is approaching and with it, the rate of change of technology is increasing. In the midst of this changing environment, higher education must produce graduates who meet the needs of business and industry.

Purpose of the Study

The purpose of this study was to determine what IS managers perceive to be the environment and the skills needed for success in that environment for future managers. This information can then be used to develop curriculum to satisfy these needs.

Hypotheses to be Tested

H_0: There will be no significant difference in how different levels of management view the future environment or the skills needed for management success.

H_1: There will be no significant difference in how managers in different types of firms, as measured by Standard Industrial Codes (SIC), view the future environment or the skills needed for management success.

H_2: There will be no significant difference in how managers in different sizes of firms, as measured by size of IS department, view the future environment or the skills needed for management success.

DATA COLLECTION PROCEDURES

The accessible population for this study was the attendees at a Management Information Systems conference hosting over four hundred conferees from business and industry and the academic community. Only business and industry IS managers completed the survey with an estimated 60 percent of those in attendance responding. Respondents were self-selecting and only 119 of the questionnaires were complete and usable.

A synopsis of the respondents are shown in Table 1.

Firms were categorized by SIC. The majority of the respondents (52 percent) were from the service industry. The other SICs did not have sufficient numbers to be useful in a statistical analysis. Therefore, the remaining respondents who did indicate the type of firm were collapsed into a single cell for analysis.

The questionnaire consisted of a total of 30 questions, 29 of which were to be ranked using a Likert-like scale. The 30-item questionnaire asked managers to select computer literacy skills required of all entry-to-middle managers. The results are shown in Tables 2 and 3.

Table 1

Demographic Data on Respondents

Title or position:

	Frequency	Percent	Valid Percent
Top Management	16	12	13.4
Middle Management	56	41	47.1
Non Managers	35	26	29.4
Other	12	9	10.1
No Response	17	12	
TOTAL =	136	100	100.0

Type of firm:

Service industry (government, university, etc)	71	52	60.2
Others	47	35	39.8
No Response	18	13	
TOTAL =	136	100	100.0

Size of IS department:

Less than 100	67	49	61.5
100 to 250	17	13	15.6
Over 250	25	18	22.9
No response	27	20	
TOTAL =	136	100	100.0

DATA ANALYSIS

The data were analyzed using the SPSS/PC+ statistical package for the IBM microcomputer. A one-way ANOVA was used with a least square deviation mean comparisons to test the effect of the size of the information systems department within the firm. Since this is essentially the same as multiple t-tests, some of the data may show significance by chance. The type of firms variable was collapsed into two groups and t-tests were used to compare groups. Levene's test was used to test the variance of the resulting groups by pooling the variance for a more accurate measurement. However, there were no significant differences found on either the variables size of the firm or type of firm.

A one-way ANOVA was used with a least square deviation mean comparisons to test the effect of different levels of managers responding. Statistically significant differences at the .05 alpha level were found on three questions. Practical significance is discussed in the findings.

FINDINGS

The respondents were very homogenous on their responses. The only differences were found on three statements (#12, #17, and #21), by levels of management, as shown in Table 2. Several statements were inversely correlated to check the consistency of the responses. It should be noted that although the Likert-scale went to 6 (will not occur), none of the respondents used the 6 ranking and very few used the 5 (highly unlikely to occur) ranking.

Time has probably taught us that it is ill-advised to bet that anything won't occur in technology. Therefore, the highest averaged ranking was 3.98 (probably will not occur). The statements will be grouped, and the findings discussed, by their impact on input/output, hardware, software, data, organization/architecture, personnel, and education.

Input/Output Items

Items #8, #19, #22, #24, and #26 all deal with methods of inputting data. Voice input and output was seen as highly likely to occur (2.07) but not seen as replacing keyboards and the mouse (3.12). Item #22 is also interesting because of the significant differences found in the responses. Top management rated voice input much higher (2.69, probably will occur) than did non managers (3.34) and other business personnel (3.75, probably will not occur). Optical character readers were seen as an important input device (item #19, 2.91) and the keyboard will remain as an important input device (items #24 and 26).

Hardware Items

Items #9, #12, #13, and #16 all deal with hardware. Items #9 and #13 suggest that workers will be using multimedia workstations in a distributed environment (2.16). Microcomputers with increased speed and power will continue to replace mainframe computers (2.28). Item #12

Table 2

Questionnaire for IS in the Year 2001 Study

KEY:
1 = Certain/occur 4 = Probably not occur
2 = Highly likely/occur 5 = Highly unlikely
3 = Probably will occur 6 = Will not occur

1. People will be our most important resource. 1.58

2. Continued education/training will become even more important for information workers in the future. 1.64

3. Computer systems throughout the organization will be networked regardless of geographic boundaries. 1.71

4. Software will automatically detect grammar, punctuation, and spelling errors. 1.91

5. Top management will use integrated computer workstations. 2.01

6. English text will be capable of automatic translation into selected foreign languages. 2.01

7. Telecommunications will link information processing personnel to work at home. 2.05

8. Microcomputers will have voice input and output. 2.07

9. Multimedia workstations will be an important resource for information workers. 2.16

10. Electronic mail will replace paper for internal communication within the firm. 2.16

11. Centralized corporate databases will be available to all information/knowledge workers. 2.24

12. Optical disks will replace magnetic media for storage of permanent records. 2.26

TOP MANAGEMENT	2.75
MIDDLE MANAGEMENT	2.05
NON MANAGERS	2.26
OTHER BUSINESS	2.83

13. Microcomputers with increased power, speed, and storage capacity will replace mainframe computers. 2.28

14. The WINDOWS operating platform will replace the DOS platform. 2.29

15. Fourth generation programming languages will replace third generation languages. 2.31

16. Impact printers will be eliminated for general office printing. 2.37

17. Flexible working hours with a required core period for all personnel will be the norm. 2.43

TOP MANAGEMENT	1.81
MIDDLE MANAGEMENT	2.66
NON MANAGERS	2.40
OTHER BUSINESS	2.25

18. Group Decision Support Systems will be used in most corporations to support decision making. 2.63

18. Optical character readers and scanners will be the primary input device. 2.91

19. Technology will be our most important resource 2.94

20. Safety and health issues connected with the use of computer terminals will be resolved. 3.05

21. Voice input will replace keyboards/mouse for entering commands to computer workstations. 3.12

TOP MANAGEMENT	2.69
MIDDLE MANAGEMENT	3.00
NON MANAGERS	3.34
OTHER BUSINESS	3.75

23. Teleconferencing and telemarketing will replace or eliminate most business travel. 3.28

24. Voice will replace the keyboard as the most common form of entering text into computers. 3.30

Table 2 (continued)

Questionnaire for IS in the Year 2001 Study

KEY:

1 = Certain/occur	4 = Probably not occur
2 = Highly likely/occur	5 = Highly unlikely
3 = Probably will occur	6 = Will not occur

25. Approximately one-third of all information workers will work at home on a corporate network.	3.37
26. Touch screens will replace keyboards for entering commands to computer workstations.	3.52
27. Information systems will create an environment with careful monitoring and control of employees.	3.60
28. User friendly software will eliminate the need for on-the-job training.	3.82
29. Paper will be eliminated.	3.97

---

discussed the use of optical disk to replace magnetic media for storage of information. There were significant differences noted, however, with lower management responding highly likely to occur (2.05) and top management and other business people responding probably will occur (2.75 and 2.83). Item # 16 suggests that impact printers will be replaced for general office printing by laser printers or more advanced technology.

Software Items

Items #4, #6, #14, and #15 all deal with software. Items #4 and #6 all were ranked highly likely to occur and discussed increased power of the information system to transcend geographical boundaries and have the ability to produce error free text in selected languages. Items #14 and #15 dealt with applications and suggest Windows will replace DOS as the operating platform of choice (2.29) and the replacement of third generation languages with fourth generation languages.

Data Items

Items #11 and #12 deal with data and data storage. Item #11 suggests that centralized databases will be available for all information workers. This agrees with Items #3, #9 and #13 which suggest a distributed environment, presumably networked and accessing centralized databases. Item #12 on optical disks for storage has already been discussed.

Organization/Architecture Items

Items #3, #5, #7, #10, #17, #18, #23, #25, and #29 discuss the impact of technology on the organization and its underlying architecture and culture. Item #3 suggests the use of networks (the "information highway") to eliminate geographical boundaries (1.71). Item #5, top management use of integrated computer workstations, was also ranked highly likely to occur (2.01). However, the respondents were more ambivalent on related issues such as the use of GDSS to support corporate decision making (item #18, 2.63) and the use of teleconferencing to eliminate business travel (item #23, 3.28). Item #7, working at home, was rated as highly likely to occur (2.05), yet teleconferencing and telemarketing received a lower ranking (item #23, 3.28) and the system was not seen as a threat for monitoring and control (item #27, 3.60). Item #10, e-mail replacing paper, was ranked as highly likely to occur (2.16), yet item #29, paper will be eliminated was ranked the lowest (3.97) as probably will not occur. Items #17 and #25 suggest a more flexible work environment, using the distributed network to link workers with their jobs (2.43, 3.37). This is further discussed in the next section.

Personnel Items

Items #1, #7, #9, #20, #21, #23, #25, and #27 dealt with information systems personnel. Item #1, people as our most important resource, received the highest ranking (1.58). This correlated well (inversely) with item #20, technology as our most important resource (2.94). Items #7, #9, #17, #23, and #25 all suggest workers using the "information highway" and sophisticated workstations in a distributed environment to work at home and reduce travel. Most were rated as probably will occur. Item #21 addressed health and safety issues and was rated as probably will occur (3.05). Item #27 on the use of information systems as a monitoring and control system was rated as probably will not occur (3.60).

Education Items

Items #2, #28, and #30 dealt with education of the workforce. Item #2, the need for continued education/training, was rated as highly likely to occur (1.64) as opposed to item #28, eliminating the need for on-the-job training, which was rated as probably will not occur (3.82). Item #30 addressed the computer literacy skills required of all entry-to-middle level managers. The results are summarized in Table 3. Of particular interest was that top skills were text editing (100%), spreadsheets (99%), database (79%), and graphics (72%). None of the other skills were selected by a majority of those responding.

The ratings for third generation programming languages was also interesting. The "C" programming language was rated as the most valuable third generation language, but only eighteen percent selected "C". Only seven percent selected COBOL as valuable, even though it is required at most business schools. Decision support systems, desktop publishing, netware, and CASE tools were all ranked much higher.

Table 3

Questionnaire Skills Inventory

SKILLS	RESPONSES	PERCENT
Text editing	119	100%
Spreadsheets	118	99%
Database	94	79%
Graphics	86	72%
Decision Spt Systems	58	49%
Systems analysis	54	45%
Desktop publishing	49	41%
Networking	48	40%
Database Management	40	34%
Systems design	32	29%
CASE tools	31	26%
Expert systems	31	26%
C	22	18%
COBOL	8	7%
PASCAL	3	2%

Conclusions and Recommendations

Conclusions

This study garnered opinions from a small group of self-selecting information systems managers. As such, it was never meant to be the definitive study for curriculum design. However, as shown in Table 4, the accessible population did consist of representatives from a diverse group of firms and, as shown in Table 1, firms of varying sizes and types. Therefore the trends indicated may be extrapolated to a much larger population.

On input/output trends, the data suggest that the keyboard will remain as an important input device for the near future. The mouse, touch screens, and optical character readers will also be important options. Total voice input appears to be further in the future, although the literature continues to promote it (Gellerman, 1994).

Hardware will continue to increase in terms of processing power and speed. This will be required to operate multi-tasking programs, graphical user interfacing, and multi-media presentations. Workstations will have digital sound and video cards, optical disk storage, and multiple 66+ Mhz processors handling 25 to 75 MIPS. High quality laser, LED and ink-jet printers that support color will replace impact printers.

In software development, software that checks grammar and other mechanics of writing will be available and in demand. Foreign language conversion will be possible and will push storage microcomputer speed and power. All software will continue to push the limits of hardware. In data storage, the increased used of optical disks will enhance the power of computers by providing virtually instant access to the entire database. Even today, "a single CD-ROM disc holds more data than more than 450 floppies" (Flynn, 1994).

In organization and architecture, a world-wide network will be established as end-users demand access to a global collection of databases. This will require a greater emphasis on standardization and compatibility, and it will allow continued growth in distributed processing throughout the networks.

The demand for trained knowledge workers will continue to grow. The work force must have even more technical and human relations skills to function in a global environment. Workers will face intermittent retraining as jobs continuing to evolve with technology.

The impact for education is enormous.

Education must produce graduates, at all levels, with a complete set of business, technical and human relations skills. Further, more emphasis must be placed on critical thinking and problem solving skills that are applicable to a rapidly changing global environment. The key to success in the work place will be a solid educational foundation upon which to build and the ability to adapt to the new demands of the business environment.

Accurate predictions about technology are difficult. However, it can be concluded from this study that information systems curricula should emphasize people learning computer applications, networking and telecommunications.

Table 4

Partial Listing of Corporations Represented at the Conference

Business and Industry

AT&T
AMERICA FIRST CREDIT UNION
BLUE CROSS\BLUE SHIELD
BOURNS NETWORKS, INC
CITIBANK
COOPERS & LYBRAND
EVANS & SUTHERLAND
FIRST SECURITY CORP
GRUMMAN CORP
HELEN CURTIS INDUSTRIES
HERCULES
I.R.S.
J.C. PENNEY
KPMG PETE MARWICK
MICROSOFT
MITSUBISHI
MOORE BUSINESS FORMS
NCR CORP
NORTHWEST PIPELINE
TRW
TEXAS INSTRUMENTS
THIOKOL
US BANKCORP
WESLO INC

Table 4 (continued)

Government and Academia

DEFENSE DEPOT OGDEN
DEPT OF VETERAN AFFAIRS
HILL AIR FORCE BASE
RICKS COLLEGE
SALT LAKE COMMUNITY COLLEGE
SNOW COLLEGE
STATE OF UTAH
UNIVERSITY OF UTAH
UTAH STATE UNIVERSITY
UNIVERSITY OF MICHIGAN
WEBER STATE UNIVERSITY
WESTMINSTER COLLEGE

Recommendations

Recommend that educators continue to elicit the perceptions of business to assist us in building a stronger curriculum. Education and business must forge a strong alliance in such areas as advisory boards to provide input and insight into the needs of business, exchange programs between teachers and business personnel, cooperative work experiences for students, visitation and demonstrations in both classroom settings and in business, and financial and technical support.

Recommend that follow-up studies be done to determine when, and what, changes are taking place in the workplace.

References

Flynn, M. (1994, 14 February). The little disc with a big appetite. US News and World Report.

Gellerman, E. (1994, March). Speech-recognition technology: The final frontier. T. H. E. Journal, Vol 21, No 8, pp 14-18.

TECHNOLOGY MAY CHANGE, BUT PROBLEMS REMAIN THE SAME

Nilakantan Nagarajan
Associate Professor - Management
Wesley College
Dover, Delaware 19901

ABSTRACT

Over the past three decades, computers have been increasingly used for developing information systems for managerial use in control and decision-making. While vast strides have been made in the technological advance and innovation in the computer field, problems relating to and associated with human aspects in computer-based information systems remain the same as before.

A survey was conducted in 1979 among FORTUNE 1000 firms to identify 17 problems facing proper development and implementation of computer information systems. The author conducted a survey in 1989 among the DP executives and professionals in the Northeast, and found that the major problems were almost identical to those identified a decade ago!

BACKGROUND

The art of Information Systems Development has, over the past decade, undergone several significant changes in the methods of approach, analysis, design and implementation. The traditional methodologies and life cycle concept have yielded to the more structured ones. The centralized approach of systems development has now been, at least to some extent, replaced or supplemented by the decentralized end-user computing, thanks to the popularity of personal computers and communication advances.

More and more technological innovations and automated tools and techniques have made the life of an Information Systems professional a veritable hotbed of ever-changing demands and expectations. The traditional profile of systems analyst as an all-rounder embracing the divergent disciplines still remains intact, but only to a higher degree and more evasive and indefinite extent. Recent survey of the COMPUTERWORLD of chief executives in Fortune 500 organizations has revealed that the essential role of Information Systems area has been duly recognized for its valued contribution to the corporate strategy and continued growth in a competitive global marketplace. But, have the problems inherent in the systems development and implementation changed because of the metamorphosis in the arena of Information Systems?

A study was conducted in 1979 by Prof. Michael J.Cerullo on the problems hindering the smooth and successful systems development of computer-based information systems.(3)

A questionnaire was sent to the MIS Director of each firm, randomly selected out of Fortune 1000 firms, asking him to rank each problem listed therein in the scale 1-9. MINOR problems were ranked 1-3, MODERATE problems 4-6, and MAJOR ones 7-9. Responses were analyzed by larger group of Fortune 500, next group of 500, and all the firms. The results are presented in column 1 of the Table, in the order of categorization in Cerullo study.

Description of the Follow-up Project:

A follow-up study was conducted in the Fall '89 among the DPMA members in the Northeast regional chapters, holding positions in the Fortune 1000 firms or performing consultancy services in MIS.

A copy of the questionnaire sent for feedback is appended for reference. The 17 problems identified in the earlier study were included, but in a scrambled sequence, together with 7 more problems for ranking by respondents on a scale of 1-5 of increasing importance. The respondents were also asked to add any other issue with ranking, as they deemed necessary.

About 180 questionnaires were distributed through the good offices of DPMA chapter in Stamford, Connecticut among their members. Copies of the survey instrument were also sent to other regional chapter chairpersons of DPMA in the Northeast for publicizing the survey among their members for participation in the project. 103 valid responses were received, reflecting the interest among the MIS practitioners in the important topic in their profession today.

The rank and mean score of each problem are indicated in the columns 2 and 3 of the Comparative Table, with key indices.

The results are all the more revealing, as the top priority concerns in the systems development still remain the same, as they were a decade ago.

The more they change, the more the problems remain the same!

Discussion of the Results

MAJOR PROBLEMS

a. Lack of top management involvement

This concern remains topmost still, despite the strides of progress made in the information systems technology and tools to enable top executives recognize the value of Information Systems in their responsibilities, and use them for their purposes, overcoming the computerphobia of many executives.

The same concern has been voiced in other recent surveys published in Computerworld and CIO magazines among others.

> "Most IS professionals say information systems is regarded as a critical part of their organizations, but that it is insufficiently staffed and ill funded".(11)

While 59% of the respondents agree that their management viewed IS to be of strategic value to their firms, only 41% felt that MIS Dept. was adequately staffed, and only 30% felt that they had adequate budget allocations for their projects.

> ...For the past three decades, information technology has been struggling to make the transformation from necessary evil to strategic weapon. CIOs, consultants, journalists and business school professors can make pronouncements about the success of this transformation until they are blue in the face, but the obvious truth is that none of this can happen unless it first happens in the eyes of the CEO.(7) (Underscore supplied.)

The study of the attitudes of non-IS executives conducted by Coopers and Lybrand in July 89 by telephone interviews with 200 CEOs,CFOs and COOs is discussed by Allan E. Alter in his article in CIO, which reveals some surprising insights on the role and future of Information Technology in business.

> "...IS is going to need significant cooperation from the CEO and COO to effect organizational change. You are selling [IT] at a very senior level,[where there is]a lot more turf protection [to overcome]." (1)

b. Inadequate Pre-systems Planning and Inadequate Strategy to extend MIS to other areas of the Organization

The two issues were ranked 3 and 5 respectively in the '79 study, but now they are both tied for second place, again stressing the need to implement the precept into practice.

As reported in Computerworld, a survey of 350 CEOs conducted by Andersen Consulting among the high-tech firms in the Northeast revealed that a) fewer than half engaged in IS planning as part of their overall strategic business planning process, and b) only 11% felt that strategic planning was effective, and 20% said that it was not at all effective!(5)

Top management must recognize information as a valuable asset on par with other types of assets of the organization. There seems to be a continuing lack of prestige for the Information function, even though some inroads are being reported recently in the boardroom and decision-making structure.

In the coming years, Information Technology professionals will hopefully be accepted as part of the mainstream of business and play the role of facilitators and integrators of disparate elements and segments of their organization in a coherent and cohesive manner.

> "IS will not determine the shape of the 90's organization, ...,even though...IT practitioners will, in a very real sense, be the architects of the future. But it will be the future that business professionals want." (2)

c. Lack of Leadership and Integrating Computer with the rest of the organization

The second issue was ranked 7th in the earlier study, and is tied with leadership issue for the fourth rank in the current study. It is concomitant with the issue of inadequate strategy to amalgamate the IS plans and projects encompassing the entire enterprise. This new awareness of systems-oriented strategic approach to utilize information systems for the overall organizzation can be seen as the force behind the upgrading of this salient issue from seventh to fourth rank.

Lack of leadership and far-sighted enterpreneurial effort for use of information as a strategic weapon and effective tool for multiple purposes of improved productivity, facing the global challenges in quality, better motivation of human resources and similar factors in a modern enterprise are being hotly discussed nowadays.

In their book 21st Century Report, Prof.Hambrick and Prof. Fredrickson predict that the next century's CEO will have a vision about the company's strategic position in a global environment, and be a leader, not a boss.(6)

Thomas Watson Jr. expressed uncertainty on this nation's willingness to make sacrifices necessary to remain the greatest nation in the world anymore; he also expressed that motivated personnel are the most important asset of any business.(4)

Gardner, in his fine analysis and practical recommendations on leadership,calls for a need that 'leaders must understand the needs of the people they work with, and their larger needs for a sense of community and mutual trust, for recognition and respect, and for new challenges and visions'.(8)

In the final analysis, a true leader must combine the capabilities to show support, provide direction, encourage active participation, and enthuse followers in their pursuit of excellence. As the several articles and surveys indicate, it is now a challenge as well as a rare opportunity for IS professionals and executives to show their skills as leaders with creativity, open-mindedness and enthusiasm in a changing technology scene, rather than just remain technocrats in the background.

The issue of leadership was not even included in the '79 study, but has now been ranked high as a major problem, thus reflecting the importance of leadership for future strategic role of information systems and technology.

d. Cost and complexity of using DBMS, and Availability of qualified EDP staff

These two concerns ranked 2nd and 4th in '79 are still considered major issues, but to a lesser degree, as shown by their rankings of 8 and 7 in the current survey.

With the proliferation of software and their use in many types and configurations of computer installations, data base systems have become thorny issues, especially in SQL features and transmission methods, due to lack of uniform standards and protocols. With the changing technology, the problems become more protracted, thus justifying the concern of major degree.

As regards the quality and level of training of EDP staff, the scene is far improved from the earlier study. Still, several top executives feel strongly that colleges and universities do not meet their obligations to equip their graduates with the skills needed for the business world and marketplace.

As the technology changes, and the expectations of top brass of the role of IS Department change, the problem of finding personnel with adequate skills and training becomes acute.

A recent study by a bipartisan commission of academic, business and labor representatives has concluded that the U.S. is on a de-skilling binge for the sake of short-term growth of productivity that could prove disastrous for business and economy in the long run. The transition from school to work in the U.S. is noted to be the worst in industrialized world.(9)

While the levels of computer literacy and EDP skills have vastly improved among the IS professionals, the problem still remains of proper matching of available skills and experience to the needs of jobs to be filled. With proper planning of IS needs and training schemes for staffing the organization, this immense problem could be tackled to a satisfactory degree.

The problems felt by lack of adequate preparedness of new graduates entering real world business, as expressed by executives as well as the students are being reported with regular frequency. The results of a recent survey of 94 executives in Hartford. CT region on their expectations of entry-level MIS employees strongly supported the need for better communication skills, and knowledge of structured systems development methods and emerging programming languages like C and Pascal, and operating systems like Unix and productivity tools. The executives do not expect entering IS employees to be adept in advanced fields as AI, Expert Systems, Simulation etc. (10)

This study and various other similar studies on the need for better interaction and positive correlation between industry and campus should serve as eye-openers for better planning on long term basis, as again evidenced by the findings of the Commission on the Skills of the American Workforce.(9)

e. Management Resistance to MIS

This issue, ranked 13th in Cerullo study has shot up to rank 6 in the recent study, again supporting the premier issue of top management attitude, discussed earlier. With proper pre-systems planning, integration of IS plans across the whole organization, and leadership to motivate the entire machinery to progress towards the common goal, the question of resistance to MIS among managers could be amicably resolved. All the fears and frustrations in the different levels of organization tiers are caused by basic human factors, and it is essential to work on the resistance problem before venturing to implement any system.

MODERATE PROBLEMS

a. Cost of recruiting and retraining qualified personnel

This problem was not included in the '79 study, but now is ranked closely after the issue of 'Availability of qualified EDP staff'. The changing technology and the flood of computer products entering the market with rapid frequency cause serious concerns for an organization to keep pace with technology and competition, at the same time keeping costs under control and productivity above margin. These are human-oriented problems to be addressed with sensitivity and humane approach to get the reciprocal responsive loyalty and service from the IS professionals. Large turnover of qualified personnel without retention plans for motivated staff will only worsen the plight of an organization, if one prefers low-paid personnel for short-term cost reduction at the expense of better-paid, experienced personnel who could contribute better for that organization.

Late Bartlett Giametti, the past president of Yale and National Baseball League, likened the management of both those institutions as almost identical! If there are no proper plans for recruitment, retention and retraining of personnel in any type of organization to meet the requisites of changing technology - be it industry, government or academic campus - the long-term ill-effects for the organization and the society will be acute indeed.

The warning message should alert the powers that be to take immediate proper plans in utmost sincerity, as pointed out earlier. Otherwise, Giametti's words of wisdom will prevail! With the revolving-door policy of trades and turnovers of staff, the team may be different, but the player remains the same, and, above all, there is no value-added productivity on the whole!

b. Role of systems staff, not clarified

This problem was rated 15th in the Cerullo study, but has been tied with the previous issue of personnel for the 9th rank. This signifies once again the importance of recognition by top management of the essential role and contribution of IS division of their organization for its overall growth and welfare.

With the various innovative ideas and approaches like End-user Computing, Information Center, Downsizing and Outsourcing, it is all the more essential that the systems staff should be aware of their status, role and expected contribution to their organization for productive and professional services.

c.Other 'Minor' Problems upgraded as 'Moderate'

Two other important issues, i.e.' Management acceptance of the computer' and 'Reliability of Vendor Support' were ranked as 'MINOR' in the earlier study with ranks of 14 and 17, but have been upgraded as 'MODERATE' with ranks of 12 and 14 now.

The earlier discussions on the hesitant attitudes of some top executives on computerized systems and unfettered proliferation of products in the marketplace with questionable quality of the service and support by the vendors justify the rankings expressed by the respondents as regards thses two problems.

As pointed out in Alter's article, 'many CEOs and COOs are using computers themselves for the first time', and ' as a result of this hands-on exposure to technology, they may have greater appreciation and respect for the CIO.' (1)

The periodical reviews on benchmark tests conducted by many independent professional groups on the myriads of products only stress the need of utmost care and pre-planning on the part of a customer in product selection to suit one's needs. The recent trend of mergers and takeovers of service groups and failure of many computer companies only add to the woes of an unwary client!

d. Other 'MODERATE' problems

As may be evident from the table, other issues which were found to be 'Moderate' problems in systems development have now either been downgraded as 'MINOR'(keyed as @) or still retained in 'MODERATE' category but with less emphasis. The rationale for such treatment may be the improved levels od computer literacy and reduced computerphobia among the different levels of staff, and also simplified, user-oriented training and business tools available nowadays for better understanding and use.

e. Uncertainty of impact of changing technology, and Management of Computer Resources

These two problems were not listed in the '79 study, but are now being rated 12th and 16th among the 24 items, thus coming under the middle category as 'MODERATE'. These issues have the potential for becoming high-priority problems, as already being expressed by many executives both in IS and non-IS positions.

The Information Resources Management(IRM) is steadily gaining ground as an essential management concept for efficient and effective utilization of all information-related resources.

The uncertainty of the computer technology and the changing signals and scenerios of information processing, transmission and utilzation only add impetus to the underlying importance of the focus of this survey, i.e. proper recognition of the role of information and information systems, and planning for their effective utilization by managing them as all other resources.

MINOR PROBLEMS

a. Downgraded 'Moderate' Problems

As already discussed under the previous section, four issues have been relegated as MINOR problems from their MODERATE level in the '79 study.

b. Reliability of hardware

With vastly advancing technology, computer hardware poses not a major problem, as everyone acknowledges, and so,it is no surprise that the survey found that this issue is ranked last!

c. Reliability of Software and Feasibility Studies

These items were included only in this survey, and have been ranked 19th and 23rd in the list of 24 problems. Again, the quality and reliability of software products are improving, especially in response to user groups and intense competition. The IS professionals seem to minimize the role of feasibility studies, maybe due to the fact that,in most cases, these are carried out by other personnel in pre-planning stages. It may become more pronounced if, in future, IS professionals get more involved directly in such managerial functions and held accountable for the timely completion of IS projects within the allocated resources and the prescribed ROI standards.

d. Stress and other Motivational Factors

This issue of emerging controversy and relevance in all work related to computers warrants more attention and action. Several case studies and research findings have been reported on the physical and psychological impact and after-effects of working in computer-oriented environments. Recent see-saw legislative action on VDT in Suffolk county and New York City in New York and elsewhere only reflect the balancing trade-off between short-term economic reality and long-term welfare.

As new studies reveal more startling side-effects of far-reaching significance affecting human welfare, these salient aspects will gain more support for major concern and consideration, as has happened over the past three decades of computer utilization and information systems by individuals as well as large organizations.

CONCLUSION

The burden of the song seems to be that top management holds the key for the success or failure of Information Systems in any organization. This survey supports the findings of many others that technological advances may not alone accomplish wonders for better productivity and performance. The humanware should be accorded its proper recognition in the computerware in order to accomplish the optimum effectiveness.

A time series study like this one also helps to trace the relative importance of the issues involved in a matter like successful development of information systems, as a result of, and in response to changing trends in technology, management tools and approaches, and levels of skill and attitudes.

But, as this survey has also underscored, certain basic principles and criteria remain the same, despite the waves of technological innovations.

REFERENCES:

1. Alter, Allan E. 'Good News, bad news',CIO(January,'90).

2. Carlyle, Ralph. 'The Tomorrow Organization',Datamation(2-1-90)

3. Cerullo, Michael J. 'MIS: What can go wrong?',Management Accounting (April,'79)

4. Computerworld d/ 9-25-'89: 'Thomas Watson Jr. voices concern'.

5. Computerworld d/11-27-'89: 'High-tech firms:Do as they say..'.

6. Fowler,Elizabeth. 'Training 21st Century Executives',New York Times:Careers(6-20-'89)

7. Freedman, David. 'Cornerstones', CIO(January,'90)

8. Gardner, John W. On Leadership, NY: Free Press(1990)

9. Hoerr, John. 'Business shares the blame for workers' low skills', Business Week:Commentary(6-25-'90)

10. Nagarajan, Nilakantan. 'MIS Majors: Can they meet business expectations?', Computerworld (9-11-'89)

11. Rifkin, Glenn. 'View from the Top',Computerworld (4-17-'89)

12. US News and World Report (3-7-'88): '21st Century Executive'.

APPENDICES:

1. Survey Questionnaire on problems hindering MIS development
2. Table of Comparative Rankings of MIS Problems over the decade

PROBLEMS HINDERING MIS DEVELOPMENT

Please rank each issue on the scale of 1(**Very Insignificant**) through 5 (**Very Significant**).

PROBLEM	**Rank: 1**	**2**	**3**	**4**	**5**
1. Management Acceptance of Computer	1	2	3	4	5
2. Integrating Computer with the rest of the Organization	1	2	3	4	5
3. Inadequate strategy for extending MIS to other areas of the Organization	1	2	3	4	5
4. Inadequate Pre-system Planning	1	2	3	4	5
5. Lack of Top Management Involvement	1	2	3	4	5
6. Planning the jobs to automate	1	2	3	4	5
7. Management Resistance to MIS	1	2	3	4	5
8. Employee Resistance to MIS	1	2	3	4	5
9. Cost and Complexity of using Data Base Management Systems	1	2	3	4	5
10. Availability of Instructional Materials	1	2	3	4	5
11. Cost of Planned System outweighs Benefits	1	2	3	4	5
12. Availability of Qualified EDP Staff	1	2	3	4	5
13. Availability of Programs and Operating Software to run the Applications	1	2	3	4	5
14. Planned systems are too time-consuming to computerize	1	2	3	4	5
15. Role of Systems staff not clarified	1	2	3	4	5
16. Reliability and Quality of Hardware	1	2	3	4	5
17. Management of Computer Resources	1	2	3	4	5
18. Reliability and Quality of Software	1	2	3	4	5
19. Reliability and Quality of Vendor Support	1	2	3	4	5
20. Uncertainty of the impact of changing Technology	1	2	3	4	5
21. Cost of Training and Recruitment of Adequately Skilled Personnel	1	2	3	4	5
22. Stress and other Motivational factors in implementing new systems	1	2	3	4	5
23. Lack of leadership in Information Systems Management	1	2	3	4	5
24. Reliability of Feasibility Studies	1	2	3	4	5

APPENDIX 2

TABLE: COMPARATIVE RANKINGS OF MIS PROBLEMS

Problem	'79 Rank	'89 Rank	Score
MAJOR:			
Lack of top management involvement	1	1	4.16
Cost and complexity of using DBMS	2	8	3.11
Inadequate pre-systems planning	3	2	3.63
Availability of qualified EDP staff	4	7	3.21
Inadequate strategy to extend MIS to other areas of the organization	5	2	3.63
Lack of leadership in IS Management	-	4	3.39
MODERATE:			
Costs outweigh economic benefits	6	11	3.00
Integrating Computer with the rest of the organization	7 *	4	3.39
Availability of Programs and Operating Systems to run the Applications	8	21 @	2.63
Employee resistance to MIS	9	20 @	2.66
Planning what jobs to automate	10	15	2.76
Availability of instruction materials	11	17 @	2.71
Planned systems are too time-consuming	12	22 @	2.61
Management resistance to MIS	13 *	6	3.32
Cost of recruiting and retraining qualified personnel	-	9	3.08
Uncertainty of impact of changing technology	-	12	2.97
Management of computer resources	-	16	2.74
MINOR:			
Management acceptance of computer	14 #	12	2.97
Role of systems staff not clarified	15 #	9	3.08
Reliability of Hardware	16	24	2.00
Reliability of Vendor Support	17 #	14	2.79
Stress and other motivational factors in implementing new systems	-	17	2.71
Reliability of Software	-	19	2.68
Reliability of Feasibility Studies	-	23	2.58

KEYS:

- * Problem changing from 'MODERATE' to 'MAJOR'.
- # Problem changing from 'MINOR' to 'MODERATE'.
- @ Problem changing from 'MODERATE' to 'MINOR'.
- \- Problem not included in '79 study.

The Approaching Crisis in Computing Education: Enrollment, Technology, Curricula, Standards and Their Impact on Educational Institutions

Mark Hensel
Information Systems & Management Sciences
College of Business Administration
University of Texas at Arlington, Box 19437
Arlington, TX 76019

ABSTRACT

The academic computing field is changing rapidly. This paper focuses upon the problems and challenges facing a university computer information systems department in the face of declining enrollments while trying to raise its admission standards and simultaneously implementing new technological packages.

INTRODUCTION

The post-World War II eras of the 1950s, 60s and 70s were marked by three trends that had an immense impact on American life.

The first trend was an increase of people caused by the so-called Baby Boom of the post-war era. This caused a population growth that had never before been seen in America. The resulting larger population caused an increased demand for government services and support. The United States' population increased by 41% between 1960 and 1992; and total social spending by the government increased almost fivefold [3].

The second trend, which was caused by the first, was an increased demand for higher education opportunities. When the Baby Boomers became eligible to enter formal education programs in the 1960s and 70s their numbers caused an increased demand for educational institutions at all levels. They also created a special demand for college level education as a way to secure a good job and to achieve financial security.

The third trend was a shift from an industrialized society to a white collar, information society. America changed forever in 1956 when more than 50 percent of its non-farm labor force ceased to be classified as blue collar factory or manual labor workers and became white collar workers [26].

America's education and government institutions answered these demands in a variety of ways. The demands for more institutions of education at all levels resulted in the construction of more public and private schools. The demands for higher education opportunities resulted in the creation of government, private, and industry grant and loan programs to finance higher education and to encourage more people to seek a higher education as a way to get a better job.

DECLINING ENROLLMENT

The heady growth periods of the 60s, 70s and 80s that led to expansion in American higher education are ending and some new trends are emerging.

These new trends are forcing us to reevaluate many of the plans and programs of the past. Even the most elite schools such as Yale, Columbia and Stanford find themselves in a financial crunch, which may force them to reshape their plans and programs just to stay open. A reduced number of 18-year-olds entering colleges is causing schools to compete for a smaller entering student population [18, 21]. A short period of increases is approaching but it will not be as dramatic as the earlier Baby boom and will offer only temporary relief [21].

There is an even greater problem in the fields of science and technology including the field of computer information systems (CIS).

A series of studies by UCLA has shown that there was a huge growth in the numbers of entering freshman college students who were interested in computing occupations from 2.8% of entering freshmen in 1977 to a high of 8.8% in 1982. That dramatic increase was followed by a precipitous decline to a low of 2.7% in 1986 [18]. The decline has continued, and is expected to continue after a brief increase between 1995 and 1999 [3, 21].

At the same time the industry's demand for CIS professionals has grown steadily and the number of CIS graduates has declined just as steadily [12, 23, and 27].

This discrepancy between the growing demand and the falling graduation rates can lead to an ironic problem in the next few years. Industry will face a shortage of college-educated CIS graduates and academia will face a simultaneous lack of demand for their ability to educate CIS professionals. The decline in student interest is quickly approaching the point at which many institutions may be forced to reduce their course offerings in CIS and many institutions may not be able to support CIS as a separate major program [8].

Some members of industry have a different perspective about the problem of declining enrollments in the CIS field. Nick Simmons, executive director of MIS at Chrysler Corporation, agreed that the declining birth rate is reducing the numbers entering the field. As a result, he said, IS directors expect to hire fewer and fewer enrty-level applicants with technical degrees in the field. Chester Delaney, vice president and manager of systems human resources at Chase Manhattan Bank, said that the drop in enrollments has been caused because the CIS curriculum has not been very supportive. Delaney said that schools are teaching students about computer architectures and compiler design when they need to learn skills and tools such as COBOL. Delaney also said the problem is compounded because CIS students are considered to be elite, and many are hired by vendor firms because they expect to do more than maintain someone else's work in an IS department's entry-level positions [11].

Another problem is that some students are avoiding the tougher science and technology fields and majoring in easier fields to build a better resume. A 1991 study of the humanities fields at seven universities found that the generally lower grades awarded in science and technology were discouraging enrollment in those disciplines. Higher grades in the humanities were attracting students to those disciplines in order to build a better looking transcript so they could get into a better graduate school [1, 16].

This problem is compounded by a general decline in the education levels of Americans as compared to students in other countries. There was a drop of almost 80 points in the SAT (Scholastic Aptitude Test) scores between 1960 and 1992 although spending for education increased almost 225% during the same period [3]. A 1992 international test of 175,000 students in 20 countries by the Educational Testing Service showed that, in spite of spending more on education than almost any other tested country (7.5% of GNP), American students ranked close to the bottom in mathematics and science [15].

Another cause of declining enrollments may be increased competition. More colleges and universities were built in response to the increased demands of the past, so more institutions are competing for the decreasing numbers of students. UT Arlington faces a very competitive situation in an area served by ten colleges or universities and eighteen community colleges with more campuses planned or being built.

Tarrant County Junior College (TCJC), for example, is building a fourth campus in Arlington, Texas; and there are some concerns that this will drain more students away from UT Arlington [9]. The president of UT Arlington believes that active recruiting will add up to 10,000 new students at UT Arlington in the next 10 years. Some of the faculty believe that may not be possible due to the intense competition for students in the north Texas area [4].

Texas community colleges have had an annual growth of 5% for the last five years and private schools have had some growth, but public colleges and universities have had smaller and smaller enrollments [25]. UT Arlington's fall 1994 semester figures show a further decrease in enrollment to 23,982 students, a decrease of 500 students from the fall 1993 semester [13], which continued a downward trend in enrollment [6].

ADMISSION STANDARDS

It was Boom Time for the CIS field in the 1960s, 70s and 80s with more and more people seeking entry into the field. Demand for a technical education was so high at one time in the early 1980s that some schools eliminated students in special "weeding out" classes at the freshman and sophomore levels [8].

The University of Kansas even imposed strict enrollment limits in the computer science field in 1980. The faculty felt that the university's computing facilities were too crowded to support any more students, so they limited enrollment as a way to conserve their computing facilities.

The UTA College of Business Administration (COBA) started losing faculty in the 1980s because it was not getting enough money to be competitive for faculty pay and some were moving to other institutions. The college decided to hire more non-tenured faculty to cut expenses and to use the savings generated in that manner to grant pay raises to tenured faculty. They also decided to reduce undergraduate enrollment in the college of business by raising the COBA admission requirements from a 2.0 GPA to a 2.25 GPA; and to increase the better-paying graduate enrollment [7].

The effect was immediate. The college of business had a total decrease in enrollment from 1987 to 1992 of 17%. During the same period, it had an increase in graduate enrollment at the same time of 39% and an increase in doctoral student contact hours of 114%. This represented an overall decrease of 4% in total hours taught, but generated much more income for the college of business because of the higher fees paid by the state for graduate contact hours [7].

The rest of the UT Arlington faculty proposed raising the admission standards for all parts of the university based upon the successful program of the college of business. Some feel that raising the standards will also improve the university's image by making the academic standards more difficult to achieve, and by generating more time to do research due to smaller class sizes [6].

The university president has a different vision of the future and is encouraging the faculty to take another path [6]. The president wants to recruit more students and believes that raising the admission standards will frustrate his plans. Disagreements over the policies and direction of the university became so emotional that the vice president of student affairs and the dean of liberal arts resigned over this issue. The dean of the college of business was 'fired' by the president after announcing that he could not work with the president [5].

Another part of the controversy is the role the university should play in the community. Some believe that state-supported schools such as UT Arlington must emphasize teaching over research, and that those located in large Metropolitan areas must provide an educational opportunity to low-income and disadvantaged students. Some believe that it is an obligation of the state and the university to provide at least an opportunity for a college education to everyone. Others believe that many students would do better starting in a community college, and attending a four-year school later [5].

California implemented its famous California Plan in the 1970s and promised that the state's top students could go to top schools, others to lesser schools, and everyone else could go to community colleges. This plan worked in the boom times of the 1970s and 80s but California is facing deeper and deeper cuts in its higher education program and may have to eliminate some schools and some opportunities to reduce the costs of their social and academic programs [21].

Some are predicting smaller schools and a smaller system of higher education by the year 2000. Smaller schools may be appropriate given the over-building that took place for the last 20 years [21]. The UT Arlington faculty plan to reduce enrollment may therefore be appropriate for the times, but the president's goal to raise the university's enrollment may have doomed this plan [6].

The Department of Information Systems at UTA has experienced a different enrollment pattern. Although the COBA enrollment has been declining, the department's enrollment has been slowly increasing during the same period.

There are several reasons for this gradual increase, including a major change in the Dallas and Fort Worth (DFW) economy as it shifts from a defense to a non-defense industry economy; and an otherwise normal growth in one of the most viable industries in one of the most viable growth areas of the country [10].

NEW TECHNOLOGY

Academia is aware of the problem of technological obsolescence and the pressures it can exert on a department's curriculum. It is very difficult to stay current with the hardware and the software of computing, given the budgetary constraints on academic resources [14]; but keeping up with new technology is needed for a successful academic program, and necessary to prepare our students for their careers [24].

Although budget constraints play an important role, there is another factor that may limit the acquisition of new technologies. There is usually almost no incentive for a tenured faculty member to work very hard to introduce new technology in the classroom. Tenure, promotion, and pay increases at a four year college or university is usually based more upon research and writing for publication than upon teaching and classroom performance, so there is little or no reason to seek to implement new technologies into the curriculum unless they can be used for research and publication.

The UTA Department of Information Systems and Management Sciences, for example, places more weighting upon research and scholarly publications than upon both teaching and service [22]. No credit is given for tenure and promotion for introducing new technology into the curriculum.

In spite of budgetary constraints and the lack of incentives for the faculty to seek and acquire new technology, some new technology is acquired through those traditional, budgeted ways. There are other ways to acquire new technology.

One way is through industry grants and gifts. Industry wants to hire graduates with a technical degree from a curriculum that teaches and emphasizes current practices and methodologies [11].

Texas Instruments (TI), for example, donated new hardware and their Interactive Engineering Facility (IEF) development tool to UT Arlington. This gift gives TI a place to recruit graduates who are familiar with their CASE product. TI can also name UT Arlington as a place where companies can train their people to use IEF, and where they can recruit graduates familiar with the product.

Business 'partnerships' offer another opportunity. Like many other companies, TI offers a program for colleges and universities to create local area partnerships with businesses to fund acquisition of the IEF development tool at a relatively low price.

Another source of new technology gifts is student professional associations. The UTA student chapter of the DPMA, for example, acquired sold coffee and doughnuts to buy a client-server development package which was donated to the university.

MODEL CURRICULA

A model curriculum gives order and direction to a professional degree program and even helps justify the acquisition of new technology (for example, CASE tools are called for in the DPMA's IS 90 curriculum). Selecting a curriculum poses several problems and related questions.

The first problem is a perceived 'gap' between the expectations of industry and the practices of academia, according one study of academia and industry. Gordon Davis said that a vocational curriculum, which some in industry seem to prefer, is closely related to current practices; but that academic discipline has a longer focus emphasizing the direction of the field and underlying phenomena, with a general feeling that it is better to learn current practices on the job. James Martin said that a problem is that academia continues to teach the obsolete languages and methodologies of the past and ignores the technologies of the present. Michael Dertouzos, director of MIT's labs for computer science, said that the role of a university is not to just train people to be useful upon graduation; but to give them lasting fundamentals of the field so they'll beome long-term contributors and continue to educate themselves [14].

Several models or 'standardized' information systems curricula have been proposed by a variety of professional organizations such as the DPMA, the IEEE and the ACM. Similarities in these degree programs have confused the issue and blurred the lines between computer science and data processing so much that neither industry nor academia is completely satisfied with these curricula [14].

Although curricula are central to undergraduate degree programs in CIS, no single source identifies and compares the available professional degree curricula. One study found that the majority of AACSB accredited schools used the DPMA model curriculum (IS 90) which is considered to be more suited for business applications than the so-called more technical ACM degree program [19].

A 1991 study of American and Canadian schools offering a CIS degree program found that 43.5% of the respondents from 161 institutions said that their schools had selected the DPMA model. A different 24.8% used the ACM model; and 31.7% used a local hybrid of the two model curricula [17].

If the DPMA curriculum is selected, as is most often the case, the next question is what applications of the model curriculum are available as examples to help implement the DPMA model? Some examples which can be used include works by Becker and McGuire [2], by McCubbin and Mathews [19, 20], and by Wysocki [28].

CONCLUSIONS

Enrollments are declining due to fewer college-age students, and more competition for the fewer numbers of students. Enrollment in CIS programs is also declining for the same reasons and because of changing interests, poor preparation for technical fields, and lack of awareness of the rewards of the CIS field.

A reduction in size may be forced upon us by the smaller numbers of entering students. The eras of great growth are ending and we must compete to survive as a viable degree program.

The curriculum selected for our schools must be supported by the technology available, and we must seek the best and latest technology to train our students for the careers ahead of them, not just the jobs they get upon graduation.

Budget constraints will force us to seek new sources of funds for adding new technologies to our curricula. Partnerships with industry may become ever more important for us to be able to offer the technical packages needed by our students. Other ways of acquiring technology must also be explored.

REFERENCES

1. Alexander, Suzanne. "Trophy Transcript Hunters Are Finding Professors Have Become an Easy Mark, " The Wall Street Journal, April 27, 1993, pp. B1 & B10.

2. Becker, S. and E. C. McGuire. "An IS Instructional Model for Supporting the DPMA 1990 Guidelines," Journal of IS Education, Sum 1990, pp. 21-26.

3. Bennet, W. J. "Quantifying America's Decline" Wall Street Journal, March 15, 1993, p. A10.

4. Black, Lisa (1993a). "Amacher, UTA work toward crucial choices," Fort Worth Star Telegram, Feb 15, 1993, p. 3A.

5. ______ (1993b). "UTA Liberal arts dean quits," Fort Worth Star Telegram, March 13, 1993, 21A.

6. ___ (1994). "Growing pains lead to UTA workshops for exchange of ideas," Fort Worth Star Telegram, Mar 27, 1994, p.1.

7. Blasingame, Brandon. "Business to change for re-accreditation," The Shorthorn, April 21, 1993, pp. 1 and 2.

8. Cale, E. G. Jr., C. H. Mawhinney and D. R. Callaghan. "Declining Enrollments in Undergraduate CIS Degree Programs," Journal of Management Information Systems, Volume 8, Number 1, Sum 1991, pp. 167-181.

9. Carter, O.K. "TCJC bond vote worthy of notice," Fort Worth Star Telegram, Jul 21, 1993, 17A.

10. Colborn, K. "Special Report: Salary Survey," DATAMATION, August 15th, 1992, pp. 38-44.

11. Davis, L. "Can Education Meet IS Career Demands?," DATAMATION, Mar 15, 1989, 65-69.

12. Duke, P. "Jobs Go Unfilled as Fewer Students Show Interest in Computer Science," Wall Street Journal, Nov 27, 1987.

13. Dunklin, Reese. "Enrollment decrease 'concerns' Amacher," The Shorthorn, Sep 1, 1994, p. 1.

14. Hartog, Curtis. "Of Commerce and Academia," DATAMATION, September 1, 1985, pp. 68-69.

15. Knight-Ridder Tribune News Service. "U.S. students fare poorly in new international test," The Dallas Morning News February 6, 1992, p. 11A.

16. Leo, John. "On Society: A for effort. Or for showing up," U.S.News & World Report, October 18, 1993, p.22.

17. Longnecker, H.E. Jr. and D.L. Feinstein. "A Comprehensive Survey of USA and Canadian Undergraduate Programs in IS," The Journal of IS Education, Spring 1991, pp. 8-13.

18. Mawhinney, C. H., E. G. Cale, Jr., and D. R. Callaghan. "Freshmen Expectations of Information Systems Careers Versus Their Own Careers," CIS Educator Forum, Volume 2, Number 1, Spring 1990, pp. 2-8.

19. McCubbin, K. and M. D. Mathews (1993a). "Implementing the Four Year Model Curriculum: Article I," Journal of IS Education, Spring 1993.

20. ______ (1993b). "Implementing the Four Year Model Curriculum: Article II," Journal of IS Education,Summer 1993.

21. Ostling, Richard N. "Big Chill on Campus," Time Magazine, February 3, 1992, pp. 61-63.

22. Raja, M.K. "Report to the Department Chair," Department of Information Systems & Management Science, UTA, Jul 8, 1993.

23. Rochester, J.B. "The Crisis in Computer Education," Computer Update, January/February 1988.

24. Shorter, Jack D. and Robert L. Dean. "Computing in Collegiate Schools of Business: Are Mainframes & Stand-Alone Microcomputers Still Good Enough?" Journal of Systems Management, July 1994, pp. 36-40.

25. The Shorthorn, "Spring 1993 Enrollments Down," article in the UT Arlington student newspaper, February 16, 1993, p.1.

26. Toffler, Alvin. Future Shock, Bantam Books, New York, 1970.

27. Withington, F. "Only You Can Prevent an MIS Shortage," Computerworld, January 11, 1988.

28. Wysocki, Robert K. "On the DPMA Four-Year Curriculum Revisions," The CIS Educator Forum Volume 2, Number 4, Fall 1990, pp. 23-24.

Experiences with an Information Management Course at the Graduate Level in Maastricht

Anton J. van Reeken
Associate Professor of Information Management
Department of Business Administration
Limburg University, Maastricht, The Netherlands
P.O.Box 616, 6200 MD
Tel.: (+31).43.883693
Email: A.vanReeken@berfin.rulimburg.nl

Abstract

This paper reports experiences that the author and his five fellow tutors gathered in organizing an Information Management course at the Business Administration graduate level of the University in Maastricht. Although we tried to implement the Problem-Based Learning principle, we had to look for several non-standard answers in designing and giving the course. These experiences will be described and discussed.

1. Introduction

A course was needed with respect to Information Management at Limburg University in Maastricht. Information Management (=IM) "comprises planning, organization and control of information resources" in organizations (see e.g. Earl, 1989, p.24). Although there is some theory specific to the field, most of it is scattered over various sources. This is a consequence of the rapid development of the motor of this field of study: Information Technology (=IT). New technological phenomena not only present new problems in the management of such resources, but also generate new answers. Many of these answers are as yet still in the pre-theory stage, and are not well documented in a student textbook. However, we do find it important that students of Information Management should be able to apply state-of-the-art theory to questions in practical situations.

A course was requested covering Information Management subjects at a level at least comparable to the NIVRA (Netherlands Institute for Chartered Financial Auditors) specification. That is, it should include everything financial auditors should know about Information Management. In fact, the resulting course was of a slightly higher level. The course was designed for third and fourth-year students of business administration in The Netherlands, which is comparable to a first-year graduate course in the UK or the USA.

The leading educational principle at the University in Maastricht is Problem-Based Learning (see e.g. Boud, 1991). The duration of a standard course is nine weeks: seven weeks of classes, one week for revision, and one week for taking exams. During such a period of nine weeks ('a block') students take two courses. A student devotes about 20 hours per week to each course. During the seven weeks of classes, two periods of two hours each are scheduled for work in groups of about 12 students (PBL groups). The remaining hours are utilized out of class, in the library, in discussion with fellow students, preparing assignments, or at home reading and studying. Such PBL groups are 'tutored', i.e. accompanied by a staff member guiding the educational process in the group.

This paper reports the experiences of the author and his five fellow tutors in designing and presenting this Information Management course. Although we tried to implement the Problem-Based Learning (=PBL) principle, we had to look for several non-standard answers to problems resulting from organizing a new course in a field in which theory is developing rapidly. The author was course co-ordinator and tutor and designed the course in co-operation with Jos Rutten, a Chartered Accountant, who was also one of the tutors.

The structure of this paper is as follows. In the next section the four major questions we had to answer in organizing the course will be addressed. One of these questions being whether we could design the course according to the PBL principle within the constraints that we faced. In section 3 the solutions that were found to these questions, are given. Section 4 presents results from an evaluation of the (1993) course, leading to some conclusions in section 5. Section 6 lists a few references.

2. Course design questions

We experienced several questions concerning the course. To mention four major ones:

- First: How should we teach students to apply Information Management theory to questions in practical situations? What should we take as the general principle of the course: would the PBL principle be appropriate? Should students as a group define their learning goals from the problems presented and go ahead accordingly, as is the usual practice with PBL?

- Second: How should we assess students? Would the format of routinely followed PBL examinations be the right one? (The normal PBL examination is a kind of multiple choice test examining knowledge at the end of the course. It is a list of about 150 statements from the field with a choice out of three possibilities, 'true', 'false', and 'question mark'. A correct answer earns one point, a false answer receives one penalty point and 'question mark' is zero points.)

- Third: How could we achieve a high output from the course, i.e., highly motivated students and a low failure rate?

- Fourth: How should we sequence the course contents? What should be the leading principle for sequencing?

In the next section, we will present nine solutions addressing these four questions and that together offer a consistent course design.

3. The course design

In discussing these questions among ourselves, answers sometimes came more or less immediately, and sometimes through an iterative process. Some questions needed more than one answer and some answers related to more than one question. We ensured that these solutions were not in conflict with one another. We had to make some compromises. However we believe that the results were acceptable. The first question (the teaching principle) will be addressed by solutions a, b, c and d. The fourth question (the sequencing principle) by solution e. The third (how to motivate and obtain a low failure rate) is also covered by solutions b, d and e, along with solutions f, g and h. The second (the assessment method) is covered by solutions g, h, and i.

a. The question of how to teach students to apply theoretical knowledge to questions in practical situations was an easy one since teaching in this university is based on the PBL principle. Problem-Based Learning not only implies that problems motivate students to study relevant literature, but also that students are stimulated to apply the theory to the problems presented. PBL is tailor-made for such learning goals.

b. Hence, according to PBL, assignments were introduced to stimulate students to work regularly during the course, to force them to go to the required level of study, to stimulate discussion in the group and to give them a first opportunity to 'solve' problems. The type of problem and difficulty of the assignments were strongly related to the end-of-course examination (see subsection i below). About half of the problems that we offered during the course were to be anded in at the start of each group session in writing. These written contributions were assessed (see subsection g below). Assessment of PBL student work, however, is not a regular PBL element. The major function of assessment was to stimulate students in achieving educational goals. An example of such an assignment is given in Exhibit 1. All assignments, of course, were discussed according to the PBL principle, that is, both before students began the assignment and after they had given an oral summary of their results.

c. One factor that was hard to reconcile with the PBL principle was that we had to take the NIVRA specification as a minimum required level. The PBL principle allows individual differences in learning goal definition and textbooks covered by the students. We could not allow this. First: with strict PBL we could not guarantee a course at NIVRA level. With PBL one cannot be sure what literature will be studied by the student, and it is one of the NIVRA requirements, on which the study plan for the students aiming at becoming Chartered Accountants (the majority of the students following the course) is based, that the literature studied can be specified. Second: because of the rapid development of this field we wanted to exclude 'old fashioned' theory. Consequently, we had to be more precise in presenting the problems and the relevant literature. So we gave assignments made up of a mini-case and a question, and also specified the literature relevant to the assignment, as can be seen in Exhibit 1.

d. Another non-PBL element that we introduced, were lectures. At the start of the course, at one-third and two-thirds of the way through, and near the end of the course we scheduled lectures. This was partly to break the group session rhythm, partly because we wanted to give an overview, however simple, at the start of the course and a more advanced overview before the examination, and partly because two subjects (evaluating IT investments and organizational change) could be more efficiently dealt with in an oral presentation. Some lectures or parts of lectures were given by external professionals.

e. As a guiding principle for sequencing the subjects of the course, we took a line from simple organizations and simple means to complex organizations and advanced means, with two intermediate stages in between. For each of the four settings we presented planning, organization and control problems.

f. Tutors should be able to judge the answers students find to the assignments given. But because of the rapid development of information technology, we could not expect our regular economics faculty to be up to date with even the current introductory state-of-the-art literature in the information management field. Involving them as tutors (a standard PBL option in our university) would in our opinion have created more problems than would be solved. Such tutors would be unable to say why certain differences of opinion exist in the literature, should students table conflicting literature they had come across. We feared that this would demotivate the students. In order to secure a high level learning process, we opted for Information Management professionals as tutors. We felt that professionals could be trained to be tutors much more easily than regular faculty could be trained to be an IM tutor. Of course, we are lucky to have professionals available in our university, in both faculty and service departments. Finally, two professionals from the 'IT department' were ready to work with us as tutors. The other four tutors were IM professionals from the faculty.

g. In order to stimulate regular and sufficiently deep study even further, we introduced study groups of three (or two) students. We decided to have the study group, rather than individual students, hand in the assignments. The assessment related to the paper rather than to individual students. The grade which was awarded to the paper was the grade for every student in the group. The average mark of a student's assignments was the first element in the final assessment.

h. The author, having some experience with PBL, was aware that students do not always behave as the PBL principle assumes they do. Some students do not do their homework every time, even when they have committed themselves in class. Some students are present, but do not contribute to a discussion. Some students do not tell in the group what they found in the literature. Some students are ill or go on holiday during a course, etc., The solution, we thought, was to assess 'participation' separately and to produce some weighted average between the scores of 'assignments', 'participation' and the final examination. Eventually, we chose a form in which the three parts were weighted equally. Every student began with a participation score of 7 (say 70%), reduced by 1 point for every absence unless this was offset by positive participation in other meetings. There was no formal obligation to be present at the group meetings. This mark was meant to stimulate students to participate, not because we wanted to assess their participation.

i. The final examination was based on a case description in which we found many of the elements that we had introduced in the course. This case, called 'Van Foreest' (EXIN, 1989), was written by an examination institute for informatics courses in the Netherlands. They were kind enough to permit us to use it. This case, with the questions that had to be answered, was supplied to students at the last session of the course. Students could prepare their answers with other students and with all the literature at their own pace during the last week of the course. However, they had to write their own answers and, following a screening by their tutor, these were orally discussed in an individually arranged session between tutor and student. Our ultimate goal was that these students, if they were presented with questions in practice like the ones we asked in their examination, would be able to find the right answers. This goal would in our opinion be realized if students were able to score a 'satisfactory' mark in this kind of examination.

With the above nine solutions we could design and organize an Information Management course (Van Reeken, 1992). The course ran during February, March, and April of 1993. Seven groups of twelve students enrolled for the course, 84 students in total. Six students left the course before the second session. The section below presents and discusses the results of the course as reflected in the marks given to the students, the results of an evaluation of the course by the students, and the results of an evaluation by the tutors.

4. Evaluation

At the end of the course an evaluation questionnaire was distributed among all the students of the course, asking for marks and remarks on various aspects of the course. These questionnaires could be filled in at home and handed in (anonymously) at the secretariat. In total, 77 students submitted an examination paper and 68 completed forms were returned. From these answers we have selected a few aspects that will be reported below.

a. For us, as designers and tutors of the course, the marks given to students give essential information about the output of the course. A substantial number of 'insufficient' marks would indicate to us that the output of the course was too low. Had that happened we would not have been too happy to report about this course in public. We consider it desirable that every student should pass, since in our opinion that is not only why we do the course, but it is also an indication of student motivation and interest in the topic. Moreover, this be a reward for all the effort which went into the design and actual running of the course. Hence, we were interested not only in the results of the evaluation by students and by tutors, but also in the distribution of the marks these students obtained for their examination paper. Marks in our country are usually on a scale between 0 and 10 (maximum), with 6 being a pass. The distribution of marks given by all six tutors is given in Exhibit 2.

   Hence, all 77 students who submitted an examination paper succeeded, and several students obtained an outstanding result. The separate results for each of the six tutors do not give extra information. With this result we felt that we had largely achieved what we had in mind. We would even have accepted a distribution shifted one point to the left, i.e., a range from 5 to 8 with a distribution of the same shape. Consequently, we felt that the nine solutions that we had found to the four questions listed in the introduction must have been about right. Of course, there could be some improvement, and we hoped to get that information from the questionnaire.

b. Among other items. the student questionnaire evaluated the following aspects:
   - did the course fit student's knowledge at the start of the

course;
- did students enjoy doing the course;
- how did students assess the course;
- how many hours did students study;
- how did students assess the tutors.

The evaluation of these items can be seen in Exhibit 3 to 7, and is discussed in more detail below. Note that not all students responded to the questionnaire, or to every question.

* The course did not fit student's previous knowledge.
The distribution of judgements made by students (between 1=Disagree completely and 5=Agree completely) on the statement "Content-matter was adapted to my prior knowledge" is given in Exhibit 3.

About one-third of the students indicated that the course did not fit their previous knowledge. Since we were informed about this by the students after the first week of the course, this evaluation result did not come as a surprise to us at the stage. We had hoped that we had repaired this problem, but presumably we had not succeeded. This problem could be traced to inadequacies in two previous courses in which, despite previous agreements, several information technology subjects had not been dealt with. Our lesson was that we had indeed taken previous courses for granted and had not explicitely considered their contents when designing this course.

* Students enjoyed doing the course.
The distribution of marks by students (between 1=Disagree completely and 5=Agree completely) on the statement "Taken together, I've worked with enthusiasm in this course" is given in Exhibit 4.

* Students gave a high mark to the course.
The question was: "If you were asked to assess the course in the usual scale (10 is superb, 6 means sufficient, 1 is bad), what mark would you give for
a: the course in total;
b: the organization of the course;
and c: what you have learned in the course.
The marks are given in Exhibit 5.

Several students did not enjoy this course, were discontent with its organization, or felt that they had not learned enough. About half of these complaints (4, 5, and 1 of the three classes of complaint) coincide with the fit-problem mentioned above. Hence, an inadequate fit reduces enjoyment and probably also motivation. We intend to address this problem more adequately next time. There were other remarks which we will use to improve the course, but these will not be discussed in this paper.

* Most students studied normal hours for the course.
The distribution of study hours per week is given in Exhibit 6.

We had planned that this course should require about 20 hours of study every week. Many students thought to be about right. Ten students reported an excessive number of hours of study, but high numbers of hours coincided with the bad fit mentioned above for only five students. An analysis of students' remarks at the bottom of the questionnaire indicated that the texts to be studied were unevenly spread, with more set material at the start and less at the end of the course.

* Students were content with their tutors.
Students rated their tutors at the usual scale (see above). The distribution is given in Exhibit 7.

Students and tutors were content with each other. We think that this is due to the fact that we had professionals as tutors. With respect to the marks students and tutors give each other, the author has observed a resemblance between the distribution of the marks students gave to tutors and the distribution of the marks tutors gave to students. This effect has been observed by the author before. This time the marks given by the two parties could definitely not have influenced each other. The differences between the distributions is very small (Wilcoxon's test gives $\underline{u}$=0.357). There is a tutor effect, although it is small.

c. Tutors expressed their opinion collectively as well as individually.

* Tutors were content with the course as well.
Finally, we also have the opinions of the tutors on the course. In general, tutors were happy with the course as well. We asked the tutors who helped us in running the course for their opinion, both during the course and at the end of it. Four tutors, two from the faculty and the two tutors from the IT department, expressed individually to the author that they 'had learned a lot from the course'. When asked to specify this, it turned out that they all meant that they had learned professionally and with respect to the theoretical background of IM in particular. All tutors had a good time doing the course.

* An open examination improves the results.
Furthermore, all tutors were of the opinion, from examination results and from the responses of students after the oral examination, that the open final examination has some unexpected benefits. These are:

- The quality of the answers was much higher than if this examination had been timed and supervised. The results were tidily presented and complete. Some of the work handed in resembled a professional report rather than an examination result.

- Some students even improved, however marginally, our own answers for the case at hand.

- With this examination students said that they had had the opportunity to learn by answering our questions. Normally, students do not learn a lot from making a test or doing an examination. This could explain the higher quality mentioned above.

This should not be misinterpreted. A reaction such as saying that, with this method, anybody can get a good mark, would miss the point. The learning goal was that students should demonstrate that they can apply the theory to problem cases, and that is exactly what they demonstrated. Of course, we orally tested whether the students knew what they were talking about. Only in cases in which facts must be known by heart would this approach not be applicable.

We think we obtained enjoyable results, which encourages us to continue to work along this line. We summarize our conclusions in the next section.

5. Conclusions

The changes we made to PBL, partly under the pressure of circumstances, partly in order to improve the PBL system, gave positive results in the course described. This is not the first course the author has designed and co-ordinated, but it is the first in which he was not the sole tutor. Earlier courses were carried out alone, or were co-ordinated by someone else. It is also the first course in which so many changes to PBL were introduced. Earlier courses yielded results in line with these, but not so well documented and articulated.

We recommend the following changes to PBL, or at least that consideration should be given to changing the PBL system in situations like the one outlined in the introduction:

a. Giving the relevant literature with each task did no harm to the goals of the present course. On the contrary, it turned out to be efficient, since students don't waste time and don't lose motivation.

b. Participation should be assessed and weighed in the final mark. How that can be done best is still an open question for us. Some discussions arose amongst the tutors in the course of using the participation evaluation system which had been planned.

c. Groups of a few students have at least four advantages over the usual PBL setting with single students. The assignments can be heavier, hence more realistic, students learn more from each other, motivation is not an individual element, and social control works positively.

d. Having students submitting written results to assignments improves the output of the course. This also implies work for the tutor. If this aspect is properly balanced with a final examination, with respect to the work involved for tutors, the cost per unit of output (i.e., per successful student) should even decrease.

e. Tutors should be able to judge the answers students find to the assignments given. Such a tutor is also in a position to motivate students to undertake further and deeper treatment of the subjects. A tutor who is not able to judge the answers, demotivates the students: if he does not know why should they?

f. Open examinations improve the output of a course, unless the purpose of the examination is to assess what students know by heart.

g. The distribution of teaching effort over a course and the total amount of effort given to a course should be a point of consideration, especially at a school of economics. It could be said that the course required more input than average. But it also produced more output than average. When more effort produces a more than proportional output, more effort is economically justified. Effort should be measured per successful student, not per course.

6. References

Boud (1991): Boud, D., & Feletti, G. Eds.: *The challenge of problem-based learning*; Kogan Page, London (UK).

Earl (1989): Micheal J. Earl: *Management Strategies for Information Technology*; Prentice-Hall Int., UK.

EXIN (1989): Casus J. & J.L. Van Foreest N.V.; in: *Modelexamenopgaven Informatiebeleid en Informatieplanning (HS.9)*, Stichting EXIN, Amsterdam (NL).

Van Reeken (1992): A.J. van Reeken: *Blokboek Informatiemanagement (5021v)*; Vakgroep bedrijfseconomie, Rijksuniversiteit Limburg, Maastricht (NL).

Appendix with Exhibits

Exhibit 1. An example of an assignment.

> **Assignment 13**
> Mr. Ooievaar, President and CEO of Ooievaar BV, was convinced by a colleague and friend to visit a symposion on Strategic Information Systems together. He had to admit that the nearby skiing opportunities had been the final argument, but much to his surprise he had also learned some interesting theories about using information technology for competitive reasons. Unfortunately, it was not so clear to him how to apply those theories in his organization.
> He decides to form a study-team and to ask it to advise him on how Strategic Information Systems could be defined with Ooievaar BV.
> **Please, submit your advice with your argumentation as an internal note to Mr. Ooievaar.**
> Literature: Bots 23, Earl 3, Scott Morton 6, Reader (Van Reeken, Rockart, Ward)

Exhibit 2. The distribution of tutor marks to students.

mark	6	7	8	9
# of students	9	40	27	1

Exhibit 3. The distribution of judgements made by students (between 1=Disagree completely and 5=Agree completely) on the statement "Content-matter was adapted to my prior knowledge".

judgement	1	2	3	4	5
# of students	5	20	17	18	6

Exhibit 4. The distribution of marks by students (between 1=Disagree completely and 5=Agree completely) on the statement "Taken together, I've worked with enthusiasm in this course".

judgement	1	2	3	4	5
# of students	3	5	15	38	7

Exhibit 5. Marks given by students with respect to three course issues.

mark	4	5	6	7	8	9
course in total	1	7	24	28	8	-
organization	3	6	23	28	6	2
learned	-	2	22	26	16	2

Exhibit 6. The distribution of study hours per week.

hours studied	<15	15-19	20	21-24	25-29	>29
# of students	5	21	26	6	5	5

Exhibit 7. The distribution of student marks to tutors.

mark	6	7	8	9
# of students	8	33	25	2

GLOBALIZATION AND THE INFORMATION SYSTEM CURRICULUM

A. K. Aggarwal*
Bruce Rollier
IQS Department
University of Baltimore
1420 N. charles Street
Baltimore, MD 21201
*410.837.5275; aaggarwal@ubmail.ubalt.edu

ABSTRACT

Organizations are losing their identity as they cross borders and become part of several civilizations creating what many have termed a 'borderless' world. This is creating havoc for managers who are faced with new problems of changing work force, instant information access and ever changing competitive environment. What is the role of business schools in this? This paper addresses some of these concerns. The paper looks at what makes a manager 'global' and suggests skills that can be taught to meet these global challenges.

INTRODUCTION

Advances in information technology are making information available anywhere any time any place with little effort. Tasks that were not feasible just few years ago are becoming trivial. This information explosion combined with political changes in Europe and opening up of markets in third world countries is creating tremendous opportunities for some but havoc for others. Organizations, both large and small, are doing business across borders in record numbers. Distances are becoming meaningless. Researchers are talking about 'borderless world' and 'borderless products' [10,11]. In a series of articles Reich [11,12] and many others have questioned the identity of local versus global organization and local versus foreign product. For example, Boeing has formed alliances with companies in Japan, Europe and North America to produce Boeing 777; Honda Motor Co. is exporting cars and trucks from US to Japan and British Airways is planning to invest heavily in US Air.

What does this imply for managers? A managers who is information literate, technically competent, culturally elite and a rational decision maker. Many companies have referred to such managers as Global decision makers. Companies want managers to understand the need of global customer and global environment early in their careers.

What does this imply for Business Schools? If corporations are becoming global, then Business Schools must provide basic training to produce global decision makers. For educators it implies restructuring and revising curriculum and even teaching methodologies to meet the business challenges of the next century. This paper discusses MIS oriented issues involved in international environment and suggests a curriculum that develops global decisin making skills.

The paper is organized as follows: the next section discusses the literature; the following section identifies IS/IT issues involved in decision making in a global environment; the following section proposes a curriculum based on issues identified in previous two sections last section discusses general implications of this curriculum.

LITERATURE REVIEW

Because of the changing world order and interdependence of world economies there is a general thrust towards "change", be it local or global. Many researchers and professional societies are working to reflect this change. Research in this area has taken two directions: one stream emphasizes the importance of globalization in organizations and the role of information technology in its management and the second stream attempts to reflect these changes in business curriculums.

Drucker [5], Hedlund [6], Omahe [10], Reich [11], Senn [13], Stair [14], Thurow [15] and many others have provided insights into the role and structure of organizations of the next century. The common theme is globalization; and survival will depend on how effectively an organization can compete in the global environment. Keen [7] and Blakes and Jarvepaa [8] discuss the role of information technology in a global environment. According to Keen, telecommunication is the single most important IT issue and he compared it to a "...highway system with information systems being the traffic". Blakes e. al.[8] provide an overall role of IT in multinationals.

Second research stream focuses on business curriculum. Data Processing Management Association (DPMA) task force consisting of academicians and practitioners published a report [4] in 1990 based on the survey of almost 1200 colleges and universities in North America. Though they did not mentioned globalization explicitly, they recognized the importance of diversity, communication and facilitation. The report stresses the importance of IS professionals to "...interact and understand a more diverse user group...IS professionals will need ever improving communications and facilitator skills". ACM/IEEE task force consisting of computer scientists published its report [3] in 1991. The report is the result of extensive discussions of engineers and computer scientists at national conferences, specific requests to experts in various areas and the establishment of a group of 120 reviewers. The curriculum is designed for engineers and computer scientists but it does recognize the social, ethical and professional issues. AACSB in its latest report [1] on standards for business schools accredition and their curriculum has clearly defined the importance of ethical and global issues in business curriculum.

It is clear that rapid change in business environment requires a dynamic curriculum. The move toward globalization of curriculum is just being recognized. Both DPMA and ACM/IEEE curriculums currently lack the explicit global dimension. AACSB recognizes the importance of internationalization of curriculum, however, it leaves upto the individual institution to implement it. Next section provides some insights into the characterstics of a global environment.

IS/IT ISSUES IN A GLOBAL ENVIRONMENT

Information Technology (IT) is the core of global decision making. It facilitates communication and provides decision making support to individuals and groups across borders. The actual IT architecture will be a function of organization structure: global, multinational vs international; product vs geographical; or hetrachical; or some combination of above (see Bartlett and Ghoshal [2] and Manheim [9] for further discussion). However, IT architecture is not of concern here, what is relevant here is Information System (IS) related issues that are important for global decision making. Deans et. al [5] interviewed several IS managers of multinationals and identified IS related issues. Though they did not group these issues, for discussion purposes we have grouped them in four themes; communications, decision making, Data and management. Technology is the **BASE** of all themes as it enables other themes.

Communication: Nothing can be accomplished without effective communication. This is needed in team work, management and/or independent routine tasks. Individuals in different countries should be able to communicate and understand each other any time any place without interference. Due to time differences, same time same place to any time any place communication is almost a necessicity in a global environment. This requires an understanding and management of communication technology. Many such technologies are available or are in the development phase. E-mail, INTERNET, FAX and Electronic data Interchange (EDI) allow transmission of documents and messages. Global networks and its management are the critical architectural issues here. Security, standardization and interpreters are major managerial concerns.

Decision making: Decisions are made by individuals or groups. Decisions may be autonomous, collective or interdependent. What may be structured task in one country may be unstructured in another. Systems developed in a global environment are global in nature. Computerized support may vary from country to country. Systems may be constrained by lack of sophisticated technology available in that country.

Individuals are usually supported through systems like decision Support Systems (DSS), Executive Support Systems (ESS) or Expert Systems (ES). However, end user computing is very different in global environment. Managers have different decision making styles, interpretations, logical sequencing and modes. All these must be incorporated in technology supporting globalized decision making. Many global decisions will be made by ad-hoc teams with team members and group leaders changing from task to task.

Technologies like teleconferencing, Group Support Systems and videoconferencing are needed. Once again standards, interpreters and cultural understanding is required. Different decision making styles must be recognized for teams to succeed. Within all this diversity managers must make ethical and rational decisions.

Data: Data must conform to borderless environment. Who should have access to what data in what form must be resolved. Security and legal issues related to transborder data flow must be considered. Corporate versus local, centralized versus distributed and individual versus group data issues must be considered. Data ownership is an important administrative and maintenance issue. In some countries data may not reliable or may not even be available. Issue of missing and unreliable data may make global decision making even more complex. Data architecture may well be the function of organizational architecture.

Management: This is a critical issue which enables administration and implementation of other themes. A global manager must have global perspective. Any theme be it technical, global, cultural or ethical must have global perspective. Key is the recognition of **diversity**. Technology transfer, software piracy and human resource management are critical managerial issues. Intellectual property copyrights in one country may not be recognized or enforced in another country. How much technology to transfer, how to protect software and data without jeopardizing decision making effectiveness requires local vs global tradeoffs. How to manage diverse groups is an imporatnt part of human resource management. Different managerial skills may be needed to manage individuals and groups due to differences in culture, social background or computer sophistication.

Management requires rational decision making. This require critical thinking and even analytical problem solving, i.e., developing and building models and looking at alternative solutions and making rational decisions. Many analytical techniques like linear programming, simulation, heuristics, inventory modeling may be needed to survive in global competitive environment.

Themes discussed above are interdependent and not mutually exclusive. For example, a managerial decision to have enterprise-wide data base have direct impact on data, management and communication.

A CURRICULUM

Educators are feeling the challenge of globalization. Countries, organizations and managerial jobs are changing and educators must change to reflect these changes. The key question is: How can we train global managers? Educators must focus on the change agents and their implications for global managers and then relate them to themes identified above.

. How is the environment changing?
. Globalization of businesses
. Interdependence of units, functional areas and organizations
. Service orientation
. Rapidly changing information technology
. Changing work force
. What are the goals for the future?
. Solving problems using information technology in a competitive and global environment
. Using Information technology for strategic advantage
. Emphasis on problem rather than tool
. Ethics involving information technology
. Awareness of people
. Implication for Curriculum:
. Emphasis on information technology and dynamics of problem solving.
. Human resource management: awareness of diversity
. World cultures- awareness of various ethnic cultures
. Group awareness and group problem solving

We propose a curriculum to reflect issues discussed above. Skills can be grouped as core, IT related and inter-disciplinary. There are two ways of including international component in the curriculum; one is to explicitly design courses related to international components: and the second is to include an international module in each course. we propose a mix mode here. the curriculum is presented in terms of skills needed and courses that provide them.

PROPOSED CURRICULUM

BUSINESS CORE:

. Introduction to Business including international business
. Computer competency and literacy
. Foreign Language & Culture
. Analytical problem solving and global decision making
. Ethics & Laws (local and international)

IS/IT RELATED:

- Telecommunications including global network architecture
- Database Management Systems
- Distributed DataBase Systems including enterprise wide data modeling and transborder data flows
- Global Support Systems including ES/DSS/ESS
- Analysis, Design & Implementation of global systems
- Interpersonal skills and Group Decision Making
- Emerging Technologies and End-User Computing
- International Information Management

INTERDISCIPLINARY:

- Total Quality Management
- Human Resource Management in a diverse environment
- Strategic & Information Systems Planning
- Global Financial Management
- Global marketing Management
- Capstone Project with an international theme

GENERAL IMPLICATIONS

The proposed curriculum provides a starting point for training global managers. Curriculum is divided into three parts. The first part provides foundation knowledge for global managers, i.e., understanding of law, ethics, international business, different cultures and different decision making styles. These are general business requirements necessary to **understand** the nature of global business environment.

The second group provides knowledge of information technology, management and its use in decision making. Some new courses like Global support Systems, International Information Management and Group decision making will develop skills needed for negotiations, working in groups and global data manipulation. Knowledge and management of new technology and its implications will be taught through emerging technologies and International Information Management courses.

The third set of courses provide integrated business knowledge. The key to future is to be able to make decisions in an integrated environment where functional lines and geographical boundaries are becoming fuzzier and fuzzier. A global manager will not be able to make decision that impacts only his/her immediate area but will have to consider its impact on other areas. For example, a marketing manager while making his/her plans for next year will have to consider its impact on production scheduling, personnel hiring/firing (rightsizing), team building and financing. Interdisciplinary courses are essential to provide an integrated view of global organization. Many of the current courses will have to be modified to include international components. A capstone course that integrates various aspects of international business in an international setting will provide real life experience.

Curriculum presented here provides for "skills" needed to make decision in a global environment. This curriculum may have to be modified to meet individual institution's needs and strengths. The proposed curriculum is like a shell which provides capabilities but also allows for specific individual tailoring.

For business schools to compete and even to survive, they need to update their curriculum based on current and future corporate demands. The need for the next century is to train mangers for borderless organizations that are producing borderless products for a borderless market. This curriculum is an attempt in that direction.

REFERENCES

1.Achieving Quality and Continuous improvement Through Self-Evaluation And Peer Review: Standards For Business and Accounting Accrediiton, The American Assembly Of Collegiate Schools Of Business, April 1992.

2.Bartlett, Christopher A., and Ghoshal, Sumantra, "Global Strategy Management: Impact on The New Frontiers Of Strategy Research", Strategic Management Journal, 1991, vol. 12, pp5-16.

3.Computing Curricula 1991, A Summary of the ACM/IEEE-CS Joint Curriculum Task Force Report, Communications of the ACM, June 1991, vol. 34, No. 6, pp 69-84.

4.Data Processing Management Association's A DPMA: two year Model Curriculum for IS professionals, 1994.

5.Deans, P. Candace, Karwan Kirk; Goslar Martin; Ricks David; and Toyne Brian, "Identification of Key International Issues in US-Based Multinational Corporations," Journal of MIS, 1991, pp27-50.

6.Hedlund, G., and Rolander, D., "Action in Hetarchies-New Approaches to Managing the MNC". In Bartlett and Hedlund (Eds.), Managing the Global Firm, 1990, pp 15-46.

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

8.Ives, Blake and Jarvenpaa, Sirkka, "Application of Global Information Technology Issues", MIS Quarterly, March 1991, pp33-49.

9.Manheim, Marvin, "Global Information Technology: Issues and Opportunities", International Information Systems, vol 1, no. 1, January 1992, pp 38-67.

10.Ohame, Kenichi, "Managing a Borderless World", Harvard Business Reviews, May-June, 1989, pp152-161.

11.Reich, Robert B., "Who Is US?", Harvard Business Review, January-February 1990, pp53-64.

12.Reich, Robert B., "Who Is Them?", Harvard Business Review, March-April, 1991, pp 85-77-87.

13.Senn, James A., "Globalization Means 'Of Unknown Origin', SIM Network, Sept./Oct., 1991, pp 7-8.

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

15.Thurow, Lester, "Who Owns the Twenty-First Century," Sloan Management Review, spring 1992, pp 5-17.

16.Taylor, William, "The Logic of Global Business: An Interview With ABB's Percy Barnevik", Harvard Business review, 1991, pp91-105.

WOMEN IN COMPUTERS AND DATA PROCESSING IN THE PEOPLE'S REPUBLIC OF CHINA: SOME OBSERVATIONS

Mary Martin Koleski, Ph.D.
Associate Professor
Computer Technology Coordinator
Purdue Programs - Kokomo, IN 46904-9003

Danette M. Taggart
Information Technology Specialist
Hewlett-Packard Company
PO Box 58059, MS 54UDF
Santa Clara, CA 95052-80059

Carrie A. Ball
Senior Programmer/Analyst
AEGON USA Investment Management, Inc.
4333 Edgewood Road NE
Cedar Rapids, IA 52499

ABSTRACT

Professional women employed in the computer and data processing industry in the People's Republic of China are encountering changes in educational opportunities, career choices, current positions and promotion possibilities as well as their work and home experiences. This paper presents observations obtained through interviews at ten organizations in the PRC.

INTRODUCTION

The People's Republic of China has for the last several years had one of the fastest growing economies in the world. ("Size of Chinese Economy.....", 1993) The Chinese, as well as other populations of developing and developed countries, are well aware that computers help make the growth of modern economies and science possible. (Reich, 1991) Computer studies and the computer industry were virtually obliterated for over a decade during the Cultural Revolution. Consequently, the major advances in the field have been made since the late nineteen seventies. The rush to gain knowledge and prestige in an Information Age is the basis of China's interest in sending their students, academics and practicing professionals to other countries to study computers and engaging in cultural exchanges with foreign experts in the computer field. The government also has proclaimed a High Technology Plan, which gives support and status to all rapid advances made in technological areas of endeavor.

By law, women have equality with men in the People's Republic of China and have played a significant role in China's economic and scientific growth in the past 40 years. (Liu, 1992) When Chinese computer specialists were asked by the authors to estimate women's participation in the field of computer development and applications in their country, the answers ranged from 20% to 50%. The professional and domestic issues that these women face in the People's Republic were familiar to their American visitors despite cultural and economic differences between the two societies.

The topics in this presentation include:

educational opportunities, careers, current positions, promotion possibilities, and women's work and home lives. The perspectives presented by the authors are based on brief encounters with Chinese professionals, both female and male, in the computer field. This was the first time that most of these women had discussed their life circumstances and reactions to them with non-Chinese visitors.

BACKGROUND

The authors were members of a delegation of sixteen American women and one man representing various aspects of computer and data processing professions in the United States that traveled to three cities in the People's Republic of China to meet with representatives of various universities, technical centers and companies. American arrangements for the conference were made by the Citizen Ambassadors Program of People to People, a not-for-profit American organization that specializes in international exchanges. The People's Republic arrangements were handled by the China Association for Science and Technology (CAST), the official government agency which facilitates such cultural exchanges with professional groups from other nations. The visit was made during a period of almost three weeks in the spring of 1993.

The computer sites visited are considered to be among the most prestigious in the People's Republic of China. The following centers were visited:

Beijing

- The Institute of Computing Technology (ICT) of the Chinese Academy of Science
- China National Computer Software and Technology Service Corporation (CS&S)

Wuhan

- Wuhan University
- Hubei Computer Federation
- Huazhong University of Science and Technology

Guangzhou (formerly Canton)

- Guangdong Society of Science and Technology
- Guangdong Branch of the Bank of China's Financial Center
- China-Guangzhou International Computer and Electronics Expo Center
- Beijing-Guangdong Chinese Computer Center
- Zhongshan (Sun Yatsen) University

This paper represents some of the observations of Chinese professionals and American delegates as they relate to issues concerning women. It is not intended to be a scientific study and no statistical data was collected. At this stage of development in the People's Republic of China, information concerning such issues is not collected, or if collected, is not readily available. If such information exists, neither the delegates, our Chinese counterparts nor CAST officials were aware of it.

The authors attempted to collect data in as systematic a fashion as possible under the circumstances. Comparable questions were asked in each conference meeting, in small group breakdowns, and in interactions with individual Chinese participants to at least increase the validity of observations. (Denizen, 1988) Many of the Chinese participants spoke English, some well and some with difficulty, since the study of at least one foreign language is required in secondary and post-secondary education. The American delegates were ably assisted by English speaking Chinese representatives of CAST. Responses were pooled by the authors since all were not present for each of the individual exchanges. Statistical data presented are estimates offered by the Chinese participants.

EDUCATIONAL OPPORTUNITIES

All students in China must take national examinations in order to enter a post-secondary institution of education. The authors were told that a student may apply to five or six universities and will be considered by those universities according to their scores on the national exams. Acceptance by the university and by particular departments within the university was stated to be based upon scores rather than on gender. The estimates we heard were that 20% to 50% of the students in academic programs in Computer Science were female. The estimates varied from university to university. The universities in Beijing, the national capital in the north, claimed the highest percentages. The universities in Wuhan, a heavily industrialized city approximately 1,000 miles south of Beijing, had lower percentages of female students in Computer Science. Guangzhou, 700 miles south of Wuhan, adjacent to the free enterprise zones of the People's Republic and close to Hong Kong, estimated the lowest rates with female enrollment in such programs at twenty percent, while one third of the teaching and research faculty were female.

At each site visited, the number of personnel with Masters and Doctoral degrees was proudly announced. Many of these advanced degrees had been earned in American universities.

When computer professionals were asked how students were encouraged to enter the computer field, the response was that students encountered computers in school, usually beginning in high school. However, several students may have to crowd around a single computer for learning purposes. It was reported that a growing number of "middle families" have home computers, although these were said to be mostly for playing games. The women who responded to this question commented that often the families, and particularly the mothers, encouraged their daughters to enter the computer field because of the pay and prestige available. It was seen as a "nice job for girls." The working atmosphere is clean and "well suited to characteristics of girls." Respondents further emphasized that since development and control of information flow in a technological age depends on intellectual ability rather than the muscle power required in an industrial age, women have a greater opportunity to be accepted on an equal basis with men for their cerebral skills in relation to the performance requirements of the job.

In Wuhan, a computer professional stated that women could go to school at night and on Sundays to earn an education while holding full-time jobs during the week. However, the male head of the Computer Science Department at Wuhan University did not feel that part-time students could give enough time to study and laboratory assignments to become effective computer professionals.

The authors also were told that some students who received an undergraduate degree in Mathematics were "assigned" to graduate programs in Computer Science. The surmise was that those governmental officials responsible for educational policy made such decisions and passed these along to be implemented by the universities. In these instances, the gender of the graduate did not seem to make a difference.

National policy concerning the financing of education is changing for all students. In the past, higher education was free to all who could qualify through the national examinations. The Chinese government is commencing a new approach. Some students now have to take loans from their work units ("dan wei") to pay for their higher education. These loans must be repaid once the graduates find work.

Other financial constraints are being placed on the universities as well. The authors were told that subscriptions to professional journals were "too expensive" for most schools to afford. When asked how they learned new software packages, they stated that for the most part they had to teach themselves. Training was available only at the China National Computer Software and Technology Service Corporation as an integral part of their business -- to train trainers from around the country for their customers. At one university, the American delegates were told (not in an open forum but in one-on-one sessions) that faculty members are being encouraged to take additional jobs outside the university. These were not meant to maintain their skills at "state-of-the-art" level, but to supplement their incomes. However, a young, single faculty member who lived frugally at home said she preferred to read more books in all fields than do outside work. Others stated that this meant they would have less time to prepare their lectures or be available on campus for their students.

CAREERS

The Chinese professionals hold a number of different positions. Because the delegates visited a number of academic centers, they met with faculty members, from Assistant Instructors to Senior Professors who held Masters and Doctors degrees. Most sessions also included a number of graduate students. At one university two of the individuals met were involved with the academic and business computing for the institution, while the majority of the professionals were involved in Research and Development of hardware and software applications.

The delegates asked teaching faculty what jobs would be available to their graduates. Many of the professionals who met with the delegates mentioned teaching, frequently at the same university at which they had been an undergraduate and graduate student. Those who enter the "job market" generally take jobs with governmental agencies or government-owned companies. The Chinese government owns and operates virtually all major companies, such as petroleum, steel and iron works, automobile manufacturing, banks, and computer companies. An increasing number find work with a "joint-venture corporation"--companies sponsored by the Chinese government as business partners with an overseas for-profit corporation from such locations as Hong Kong, the United States, Japan or Russia. Some of these joint-venture organizations for which computer graduates work are situated in the new enterprise zones established by the government for "free market" entrepreneurial activities. (These "free market" companies are not called capitalistic companies, but the emphasis on profit and productivity are comparable to what is seen in a capitalistic society.) In such zones and free market companies, the employees can earn substantially more money than if they work for government-owned companies. However, the government-owned companies provide free health coverage, low cost housing, and pension plans that those who work for free enterprises efforts must provide for themselves. These can be expensive.

Another thing that can be frightening for a Chinese university graduate about seeking a position with a joint-venture company is the whole process of a job search: writing a resume, interviewing and competing with other applicants. Apparently these skills are not necessary when individuals are assigned to jobs by the government.

At one university the delegates were told of some other restraints placed on individuals' lives:

- If one works for a government-owned company or a university, one cannot leave the job or even apply for a different job unless given a release by the top manager.

- Universities are like cities within cities. Not only do most students live on campus, but so do faculty and staff. Their children attend nursery school, elementary school and high school on campus. It is theoretically conceivable that a child could be born to faculty parents, receive all education, from pre-school through graduate degrees, on the campus, be hired to teach at the university, and live in that same enclave until or even after retirement.

- Students are anxious to study overseas. (Either because our delegation was from the United States or because it is a fact, the US was listed as the favorite location for study.)

- If husband and wife are both academics, they cannot both accept Visiting Scholar or graduate study opportunities outside the country at the same time. One must wait until the spouse returns before leaving the country.

PROMOTION POSSIBILITIES

By law in the People's Republic of China, women have equal rights with men. ("People's Republic of China.....", 1989) Men and women receive equal salaries for the same jobs, and many professionals indicated that both men and women have equal opportunity to be promoted to supervisory positions. During the last decade and one-half, a number of women have risen to significant positions in government and business life. (Cai, 1992)

However, a number of the women encountered expressed differing opinions about the status of women in the computer field. For example:

- The Chinese society had been egalitarian for many years and there were equal numbers of male and female supervisors; but some women felt that male dominance in the workplace still exists or was increasing.

- At the Guangdong branch of The Bank of (The People's Republic of) China in Guangzhou, it was reported that 60% of all employees are women. Of these women, 10% are supervisors. The percentage of supervisors who are male was not disclosed. However, of the employees who met the delegates and who represented top management of the Bank and its financial center, all but one were male.

- At several sites women expressed the opinion that it was necessary for women to be better than their male counterparts to be deemed equally competent and to be considered for promotion. If considered, their chances were "almost as good" as a man's for advancement.

- At each of the sites visited by the delegates a man was the director of computer studies, research or operations. Only at two sites did a woman rank as high as associate or assistant director.

- At only one site, the China National Computer Software and Technology Service Corporation in Beijing, was the technical meeting conducted by a woman. She was an Associate Director of the company, and "had a national reputation as a software engineer." She was in charge of approximately 1000 employees in the software development division. She told the delegates that her division preferred to hire women as software engineers because they were more "creative."

HOME LIFE

In 1979, a "one child" policy in the People's Republic of China was established in an attempt to control the growth of the population which now exceeds 1.2 billion people or one of every five people on Earth. ("People's Republic of China...", 1989) In the past, women were married at a very early age and produced many children for whom they provided care. Now, young adults are encouraged to wait until their late twenties or early thirties to get married and then delay until their marriage is "established" to request permission to try to have a child. Consequently, this provides women more time to establish and advance in a career. The almost universal availability of government sponsored day care or the presence of retired grandparents in the home to care for a child also allows women to work even while raising a child.

One faculty member from Sun Yatsen University indicated that a number of young professional women are choosing not to have a child in order to "enjoy the good life." Others indicated that women were foregoing families and even marriage in order to devote their energies to their careers.

At the eminent Institute of Computing Technology in Beijing, the female Assistant Director said that until recently everyone was assured of a job regardless of the quality of their work. The term that the Chinese use to describe this situation is called the guarantee of an "Iron Rice Bowl" to be filled by the government. Since the government has begun to change its economic orientation and has chosen to become internationally competitive, the guarantee of the "Iron Rice Bowl" is vanishing. With changes in the economic structure in the country, it is now possible for workers to be laid off since the job market is becoming more competitive. This is especially true in such fields as information processing and computer research and development where jobs require great technical skill and high individual productivity.

Although under the Communist culture, every adult has been expected to work outside the home, this individual indicated that some women were choosing to stay home because their husbands are now making enough money in joint venture and free market companies to support the family. She noted that other women were being "encouraged" to stay home so that their jobs could be given to men.

Almost all women in China work outside the home until the mandatory retirement age (55 for men and 50 for women.) The delegates met only one women who had been given permission to work past her retirement age, but she was single, had no grandchildren to tend and was "extremely valuable to the department." She was 56.

In an increasing number of instances, one member of a married couple will go to work for a foreign-owned company for a high salary, but no benefits. The other spouse will continue to work for a government-owned company (this includes universities) in order to retain benefits such as free health coverage, pension plans, and low cost housing. (Despite a massive building boom, housing is still in scarce supply.) Which spouse works for the foreign company will depend upon who can earn the higher salary. The lower paid spouse would be expected to take over the major portion of child care and housework. However, women professionals told us that, "Usually the man goes to work for the foreign company", or "It doesn't really work that way -- the woman still has most of the responsibilities at home."

The authors were told that in some families the husband does some or all of the cooking and shares in the care of the child. Yet, others reported that the mother gets up as much as an hour earlier than everyone else, cleans, does laundry and prepares breakfast (and sometimes lunches) then awakens her husband and other members of the family. Due to space problems, the family members may eat in shifts, beginning with the men, then the children and finally the women.

If the couple is not living with the in-laws, (a common practice), the husband may assume a little more responsibility for child care, which may consist of taking the child to school, and caring for the child if the wife has to work late on her studies or a research project. However, it was reported to the authors that if the child is ill, it is usually up to the mother to take her or him to the doctor.

When a Chinese woman reported that her husband was willing to share in the household and child rearing responsibilities, she and her female colleagues were usually generous in praise of him.

The Chinese women were unfamiliar with the term "Super Mom." However, when the American delegates defined the term operationally to mean a woman who was efficient and productive on the job, kept a spotless home, cooked gourmet meals, raised a bright and well behaved child and was an intelligent and loving wife, every woman knew exactly what the term meant and agreed that it was the Chinese society's expectation of them as well.

A number of the Chinese could remember that more than a decade ago, food and consumer items were scarce at stores. Shortages and rationing were always in effect. Today, the Chinese economy is making great strides and there are shortages neither of food nor most consumer products. It was reported that in some parts of the country most families have TV's and refrigerators. Many also have tape decks, VCR's and even Karaoke machines. A growing number are purchasing motorcycles to replace the ubiquitous bicycle, but no one mentioned the anticipated purchase of a car. (When one experiences the massive traffic congestion in Chinese cities with a few cars, many busses and trucks, and an unbelievably overwhelming number of bicycles, mo-peds and motorcycles, the idea of a growing number of automobiles represents a nightmare of frightening proportions.) The older professional women were familiar with the hardships of the Cultural Revolution and the meager times that followed it. The growing abundance of consumer goods has made the lives of Chinese professional women, and Chinese women in general, substantially easier than in the past.

CONCLUSIONS

One has to be cautious in drawing any firm and fast conclusions from such a limited exploration into the lives of Chinese professional women engaged in development and use of computers and data processing. The sample of respondents was small. The sample was not representative. Inquiry was pursued under difficult circumstances of limited control of time and access. Nevertheless, in the absence of any more systematic studies, the authors feel some sense of confidence that this is at least a beginning in understanding the world of work and home life of those whom they came to know and respect as highly skilled and competent professionals and women.

With appreciation for their efforts, struggles and generosity in sharing their experiences with us, we propose these tentative conclusions:

- The Chinese encountered by the delegates were warm, friendly and very open in their willingness to share their professional knowledge and their social and economic perspectives.

- The enhanced freedom that seems to be offering Chinese women the chance for new roles comes from the changing nature of their economy and the expansion of their technology. Women seem to have a greater opportunity to be accepted for their intellectual job skills on an equal basis with men particularly when such employment is embodied in an Information Age Society.

- Despite *de jure* equality with men, Chinese professional women, like American professional women, still have to contend with *de facto* interpretations of male and female roles. A substantial source of their difficulties in managing their professional and family lives stems from the persistence of traditional attitudes and views of the roles of women and men even in the midst of a changing society.

- Chinese women in the computer field are confronted with social and professional issues and dilemmas similar to those of their American counterparts. While they were not familiar with the American idiom "Super Mom", many of these women live the reality of the "Super Mom" experience.

- Many young Chinese professional women find benefits to their careers in the "one child" family policy. Others are choosing not to have children to enable themselves to devote more energy and to derive greater satisfaction from their careers.

- The Chinese economy for many is improving dramatically. Almost any consumer item that one wants is available if one has the money to pay for it.

All of what was stated in this paper is based on very limited observations. The authors feel these conclusions are significant but should be subject to the scrutiny of more observations and more systematic studies.

The authors are grateful to their Chinese hosts for sharing their views and information with us. Perhaps by learning that people are more alike than different, regardless of the differing cultures in which they are raised, the world might be a more harmonious place in which to live.

REFERENCES

Cai Hongping, "They Come Here to Divorce", Women of China,September, 1992, p. 111.

"China: Emerging Economic Powerhouse of the 21st Century", BusinessWeek, May 17, 1993, p. 54.

"Chinese Radio Talk Show Host Scores Hit", The Japan Times, May 19,1993, p. 12.

Denzin, Norman A., Sociological Methods: A Sourcebook. N.Y.:McGraw-Hill, 1988. (Third Edition).

Levy, Richard, "China Goes to Market", Dollars & Sense, April,1992, p. 135.

Liu Ying, "Women and Households in China", Women of China, August,1992, p. 120.

Suzanne Ogden, Editor. "People's Republic of China: Tensions between Modernization and Ideology", in China: Global Studies, (Third Edition), Sluice Dock, Guilford, Connecticut: Dushkin, 1989, p. 11.

Reich, Robert, The Work of Nations: Preparing Ourselves for Twentieth Century Capitalism. N.Y.: Alfred A. Knopf, 1991. p. 30.

"Size of Chinese Economy Now Matches Germany's: Pacific Rim to Stayin Fast Lane", The Japan Times, May 19, 1993, p. 9.

Teague, Joy, et.al. "Computer in The People's Republic of China", Proceedings of The Association of Computing Machinery, 1989.

ALTERNATIVE APPROACHES TO DEVELOPING BASIC SKILLS IN COMPUTER USAGE

PANELISTS

Larry W. Cornwell
Business Management and Administration Department
Bradley University
Peoria, IL

Peggy Lane
Department of Computer Science
Indiana University Purdue University Fort Wayne
Fort Wayne, IN

ABSTRACT

With the changes in AACSB requirements and the increased computer experience of college freshmen, the introductory course that teaches basic computer skills is being reevaluated at many colleges and universities. College freshmen are now entering universities with computer skills that are taught in the freshmen level introductory course. Universities need to evaluate the content of this course or the need for the existence of the course. AACSB requirements for accreditation have also changed which is forcing colleges of business to evaluate the core requirements for business major.

Several alternatives are exist to solve these problems. One alternative is the elimination of the introductory course and place the responsibility of developing the computer skills on a student entering the business program. A second alternative is the elimination of the introductory course with the offering of noncredit courses and training materials to aid the poorly prepared students. A third alternative is the restructuring of the existing basic skills course. Many other innovative alternatives exist.

This panel will present alternatives being considered at Bradley University and several other universities. After examining the alternatives presented above, the panel will open the session to the floor for a roundtable discussion.

A STUDY OF THE CAREER EXPERIENCES OF MEN AND WOMEN IN THE INFORMATION SYSTEMS FIELD

Mary Sumner and Kay Werner
Department of Management Information Systems
Southern Illinois University at Edwardsville
Edwardsville, IL 62026

ABSTRACT

The purpose of this study is to examine the impact of gender differences on the career experiences and job success of information systems professionals. The study analyzes career experiences with respect to a number of variables, including feelings of acceptance, job discretion, met expectations, sponsorship, training, career satisfaction, and organizational commitment.

The number of women participating in the computing field is still proportionately low, according to a panel of academic professionals addressing the issue of female participation in computing at a conference entitled "Scholarship on Women and Society." (McMullen, 1992). However, as women enter the workforce in increasing numbers, more and more are choosing careers in the information systems field. In addition, more and more women are completing academic programs in information systems and computer science.

Even with these recent gains in participation in the MIS field, many writers have pointed to the "glass ceiling," that prevents women from advancing beyond lower or middle-level management positions (Morrison, White, and Van Velsor, 1987). It is important to determine if the "glass ceiling" is a phenomenon influencing women's participation in careers in the information systems field, or if MIS careers provide a different kind of context and opportunity for women.

In an early study of discrimination within organizations and its effects on opportunities for minority members such as women, Kanter argued that groups such as women have low access to power and opportunity (Kanter, 1979). Without access to power, women are often restricted to lower-level assignments and excluded from informal social networks. Excluded from these opportunities, groups such as women may lower their aspiration level and may find themselves unable to influence the course of their own careers.

There is evidence in the research that women have been the object of "treatment discrimination." This translates into fewer rewards, fewer resources, and fewer opportunities. While treatment discrimination may be manifested in terms of tangible outcomes such as fewer promotions, salary increases, training opportunities, and assignments, it may also lower feelings of acceptance into the work group and limit chances to obtain social support from supervisors and peers (Ilgen and Youtz, 1986).

With these views in mind, it is interesting to examine the career experiences of women and men in the information systems field. As a growing profession, the information systems field offers many opportunities. Prior research does show differences between information systems professionals and other professionals in terms of such factors as the need for growth, social need, and need for achievement (Cougar, 1988; Cougar and Zawacki, 1980). There is also some evidence that information systems personnel have difficulty moving into managerial positions and often face limited advancement opportunities because of the technical nature of their field (Tanniru, 1983). Although the career experiences of information systems personnel have been investigated, there is almost no research which compares the experiences of men and women in information systems positions.

THE PURPOSE OF THIS STUDY

This study will address two research questions: 1. What is the impact of gender differences on the career experiences and career success of information systems professionals?
2. What skills and capabilities gained from educational preparation and work experiences are viewed as most critical to successful performance in positions in information systems, as viewed by both men and women in the field?

A FRAMEWORK FOR CAREER EXPERIENCES

This study will analyze the career experiences of information systems professionals with respect to a number of variables originally proposed by Greenhaus, et.al. (1990) and later extended by Igbaria and Wormley (1992) in their study of how race differences affect the career experiences and the success of MIS professionals. The framework for describing career experiences includes the following variables: feelings of acceptance, job discretion, met expectations, career support received from one's supervisor, participation in technical training and management development programs, career satisfaction, and organizational commitment.

Feelings of Acceptance by the Organization

Acceptance into the organization's informal networks can increase a sense of power and belonging. Feelings of acceptance in the informal network may encourage an employee to work harder and to be more effective (Woodruff, 1983). Acceptance in informal business activities with peers, subordinates, and superiors is important.

Feelings of acceptance are important to MIS professionals, who are sometimes outside the mainstream of the organization (Barkol and Martin, 1982). Women, too, sometimes encounter difficulty breaking into the organization's informal networks, particularly the "old boy" network. To be successful, women need to feel accepted; they cannot be "too macho" or "too feminine" (Morrison, et.al., 1987). They need to be "easy to be with," able to fit in by making others comfortable, and willing to provide a new kind of interactional leadership by "hosting informal social events" (Rosener, 1990).

Job Discretion

The amount of discretion an employee possesses is a factor influencing how much influence or power he or she can exercise. Job discretion is manifested in authority, decision-making power, budgetary control, and the ability to take independent action. Job discretion motivates high levels of effort, and provides decision-making skills that can enhance performance (Hackman and Oldham, 1976).

Since MIS professionals exhibit a high need for autonomy and dominance, job discretion is especially important (Couger and Zawacki, 1980). In Morrison's view, job autonomy is also a critical success factor for women. Moving into high-risk jobs and taking career risks are key strategies influencing the career success of women (Morrison, et.al., 1987).

Met Expectations

The idea of met expectations measures the extent to which an individual's initial expectations for achieving promotions, career opportunities, and salary increases, are actually met. In other words, are actual advancement opportunities different from what was expected? Met expectations are a source of career satisfaction, especially if the expected outcomes are valued by employees (Greenhaus, et.al., 1983). Unmet expectations may produce feelings of unfairness and inequity, and may cause employees to reduce their efforts (Couger, 1988; Couger and Zawacki, 1980).

Since MIS professionals have a higher need for achievement than most other professionals, their ability to realize their met expectations may be particularly important (Couger, 1990). In the case of women, the evidence that few women reach higher-level positions sends the message that talent and hard work don't pay off and creates morale and productivity issues (Morrison, et.al., 1987).

Career Support from One's Supervisor

Career support or sponsorship received from one's supervisor is clearly a motivating factor. Supervisors can provide sponsorship, acceptance, challenging assignments, visibility and constructive feedback. Sponsorship contributes to effective job performance of both men and women by increasing motivation and chances for upward mobility (Bova and Phillips, 1981; Burke, 1982).

Without question, career support is a determinant in the success of MIS professionals (Couger, 1990; Couger and Zawacki, 1980; Ferratt and Short, 1986; Turner and Baroudi, 1986). Career support, particularly mentoring, is critical for women as well. Successful women in several studies point to having the advantage of higher-level executives as supporters and advocates (Hennig and Jardim, 1977; Morrison, et.al., 1987). Women who have mentors are likely to be better paid and to have better career mobility than women without mentors (Fitt and Newton, 1981; Combs and Tolbert, 1980). Male mentors can effectively support the career development and visibility of women because they are still in higher, more influential positions (Berry, 1983).

Participation in Technical Training and Management Development

Access to training and development provides opportunities for growth that foster career success (Kanter, 1979). Training is particularly important for the development and retention of MIS employees, because rapid technological changes require the updating of technical skills (Bartol and Martin, 1982; Couger, 1980; Kirkley, 1988). MIS professionals also need business-oriented knowledge and managerial skills in order to understand the business context for information systems development and to manage projects successfully (Bartol and Martin, 1982; Kirkley, 1988).

In the case of women, training and development opportunities are equally important. The ability to polish skills and to develop state-of-the-art knowledge is critical to women's ability to succeed (Morrison, et.al., 1987).

Career Satisfaction

Career satisfaction results from challenging assignments, recognition, career support, and progress toward achieving professional goals. Since MIS professionals have a high need for achievement, opportunities for challenging work, consistent training, and salary increments are all associated with career satisfaction. In the case of women, career progress results from the opportunity to take on challenging assignments, the ability to adapt, the drive to succeed, and the ability to work collaboratively with colleagues (Morrison, et.al., 1987).

Organizational Commitment

Job satisfaction is closely related to overall organizational commitment (Baroudi, 1985; Bartol, 1983). Career satisfaction resulting from challenging opportunities, career support, training, and acceptance will enhance organizational commitment.

Organizational commitment is one of the most important factors influencing whether MIS employees leave an organization (Baroudi, 1985; Bartol, 1983). Several studies indicate that MIS professionals exhibit little loyalty to their organizations (Tanniru, 1983; Woodruff, 1980), which may be related to their desire to find highly motivating positions. Studies of the career progression of women professionals illustrate that women who achieve upward mobility demonstrate a good deal of organizational commitment over time (Morrison, et.al., 1987; Hennig and Jardim, 1977). However, complications of women's biological role (e.g. maternity, child-rearing) makes it necessary for them to balance career and family. Women who want to have the flexibility to balance their career and family obligations may be viewed by men as not having adequate commitment to the organization. This poses a dilemma for women which expects them to choose between organizational commitment and family (Schwartz, 1989).

RESEARCH METHODOLOGY

Procedure and Sample

This research is an exploratory study to determine gender differences in career experiences in the information systems field. The population surveyed included all of the graduates of a Master of Science program in Management Information Systems. The respondents to the study graduated between 1981 and 1993. There have been 102 male graduates of the M.S. in MIS program and 46 female graduates.

The procedures for conducting the study included developing a questionnaire and an interview guide. The questionnaire was sent to all M.S. in MIS graduates, and a random sample of ten graduates (five women and five men) were selected for in-depth interviewing. It was felt that the interviewing process would provide some insight into career development issues.

Development of the Questionnaire. The questionnaire was designed to provide information on the respondents' educational background, job history, salary range, age, and career experiences. Questions from a number of validated instruments were used to measure feelings of acceptance, job discretion, met expectations, career support received from one's supervisor, career satisfaction, and organizational commitment.

Interview Guide. As noted, ten of the graduates were randomly selected for in-depth interviews in order to gain further insight into career experiences in the MIS field. The interview guide included open-ended questions in each of these areas: Feelings of Acceptance, Job Discretion, Met Expectations, Career Support, Participation in Technical Training, Career Satisfaction, Organizational Commitment, and Education.

Measures

The measures used in this study were selected from a number of prior studies in which career experience variables were investigated, and all measures have been tested and found to be reliable. Open-ended questions were used to obtain the respondents' views of training opportunities and critical skills and capabilities needed for successful job performance.

Perceived Acceptance by the Organization was assessed with 12 items (e.g. "I am accepted in informal business activities with my boss") taken from the "corporate fit" scale developed by Nixon (1985a). Responses to the 12 items, each with a five-point scale from "strongly agree" to "strongly disagree" were averaged to produce a total Acceptance score.

Job Discretion was measured with nine items (e.g. "I have very little responsibility in my job") taken from a longer "job power" scale developed by Nixon (1985b). Responses to these items were averaged to produce a total Job Discretion score.

Met Expectations was measured by a four-item scale (e.g. "My rate of promotion has been much quicker than I expected it to be"). The areas of rate of promotion, salary increases, career opportunities, and training and development experiences were taken into account. The Met

Expectations scale was developed by Igbaria and Wormley (1992). Responses were averaged to produce a total Met Expectations score.

Career Support was measured by a scale developed by Greenhaus, et.al., 1990. The nine-item scale was used to measure agreement or disagreement on a five-point scale with the degree of career development support they received from their supervisor (e.g. "My supervisor takes the time to learn about my career goals and aspirations.") Responses to all nine items were averaged to produce a total Career Support score.

Career Satisfaction was measured by five items developed by Greenhaus, et.al. (1990). Individuals were asked to indicate their agreement/disagreement with each statement on a five-point scale ranging from "strongly agree" to "strongly disagree" (e.g. "I am satisfied with the progress I have made toward meeting my goals for advancement.") The five items were averaged to create a Career Satisfaction score.

Organizational Commitment was measured using a seven-item scale developed by Alutto, et.al. (1973). Each item required the respondent to indicate the probability of leaving the organization for an alternative job given increases in pay, status, and the friendliness of co-workers. A five-point scale, ranging from "1" I would definitely change to "5" I would definitely not change, was used. Responses were averaged to produce a total Organizational Commitment score.

Participation in Technical Training and Management Development programs was indicated by two items dealing with the nature and duration of training opportunities which the graduates reported. Technical training programs were defined as programs which were designed to teach specific job-related information and skills. Management development programs were defined as programs designed to teach broad managerial skills such as supervision, coaching, decision-making and strategic policy-making.

Critical Skills. Open-ended questions were designed to obtain information about the skills and capabilities gained from educational experiences which the graduates considered to be most critical to their job performance. One question asked which skills and capabilities gained from work experience were most critical for job success.

Data Analyses. The data from respondents' educational background, job history, and career skills were tabulated and presented in tables. The scores for each of the career experience variables (e.g. feelings of acceptance, job discretion, etc.) were calculated, and an overall mean for the men and for the women respondents' for each of these variables was derived. A t-test to determine if the difference between the means for the men and for the women on each of the career experience variables was statistically significant at the .05 level was then determined.

A discussion of interview findings relevant to career experiences, job success, and educational factors contributing to career success was also developed. While this analysis was based upon a limited number of respondents, it was helpful in obtaining insight into gender-related issues in career expectations and experiences.

RESULTS OF THE STUDY

The findings are organized into sections entitled Background Characteristics, Career Experiences, Training Experiences, Critical Skills, and Interview Findings.

Background Characteristics

The background characteristics reported by the 34 respondents described their education, job history, and age distribution. Of these 34 respondents, 25 were male and 9 were female. The overall response rate was 23 percent, with 25 percent of the males and 20 percent of the females responding to the survey. The low response rate of women graduates may be partly explained by the likelihood that some women have interrupted their careers in MIS because of family commitments.

Career Experiences

The career experience scores for the male and female respondents for each of the career experience variables were averaged, and a t-test to determine if the difference between the means for the men and the women was statistically significant at the .05 level was conducted. The findings showed that the difference in career experience scores for the men and the women in the sample for each of the variables studied was not statistically significant.

Table 1: Career Experiences of Men and Women

	Male	Female	t-value
Feelings of Acceptance	1.73	1.76	-.16
Job Discretion	1.93	2.25	-1.11
Met Expectations	2.93	2.81	.40
Sponsorship	2.27	1.91	1.34
Career Satisfaction	2.29	2.13	.46
Commitment	3.79	3.61	1.36

One of the first reasons why the experiences of men and women in the information systems profession may be comparable in terms of their perceptions of their career experiences has to do with the nature of the field. Men and women are recruited into the MIS profession largely because of technical knowledge, and their job performance is based upon technical competence--particularly at the lower levels.

Since the items were reverse-scored for each of the career experience variables, mean scores in the "1" and "2" range represent positive assessments. The mean scores for Acceptance (e.g. 1.73 for the men and 1.76 for the women) indicate positive feelings of "Acceptance." This may be based upon the fact that expectations for technical competence are relatively straightforward, and that both men and women are on an "even playing field" with respect to their ability to demonstrate technical expertise as members of a project team.

The mean scores for Job Discretion (e.g. 1.93 for the men and 2.25 for the women) represent a positive viewpoint. The respondents' view of Job Discretion may be based upon the nature of the work itself, since systems work is characterized by chances for problem-solving and analysis.

The mean scores for the variable Met Expectations suggest that career opportunities, promotions, and salary increases are viewed less optimistically. There may be slight frustration with advancement opportunities because corporate downsizing is decreasing opportunities for upward mobility. Some information systems professionals may reach a technical plateau and not find opportunities to advance to management.

In the area of Career Support, both men and women seemed to indicate that sponsorship was a relatively positive element in their career experiences, with the women's view of sponsorship slightly more positive than the men's. This factor may be influenced by the predominance of project teamwork as a strategy for managing systems development projects. On a team, all members have a relatively equal footing and can find opportunities for sponsorship. Women who demonstrate technical expertise become valued contributors.

The issue of organizational commitment was less positive, with both the men and the women expressing willingness to leave their current positions for increases in status and pay. The overall uncertainty in the profession, combined with the search for opportunities which might provide greater upward mobility, may be influencing this lesser commitment. This is consistent with studies that indicate that MIS professionals may willingly leave an organization to find positions which are highly motivating and which offer growth potential (Tanniru, 1983).

In terms of overall Career Satisfaction, the respondents expressed a neutral view. A number of the factors which may be related to this view, including challenging opportunities, career support, expectations, and the changing nature of the profession, will become more apparent in the discussion of the interview findings.

Technical Training and Management Development

In their responses about training opportunities, both the men and women respondents indicated that they participated in technical training and management development programs. Technical training was more frequent, with 22 of the 36 respondents (61 percent) participating, compared to management development, with 11 of the 36 (31 percent) reporting that they participated in these types of programs. Roughly the same percentages of women and men participated in both types of training programs.

Most of the technical training programs were hardware/software specific, including: Novell networking, fourth generation tools, database administration, data dictionary, and other technical topics. In contrast, management development topics included: supervisory skills, making meetings work, leadership education, effective presentation skills, and employee evaluation.

Critical Skills

Each of the respondents described the three skills which they felt were most critical to their success. As you can see from Table 2, effective communications, teamwork, analytical skills, and programming skills were considered critical to job success.

Table 2: Critical Skills

	Males	Females	Total
Communications	.60	.89	.68
Analytical	.56	.33	.50
Teamwork	.48	.11	..38
Programming	.44	.22	.38
Organizing	.36	.11	.29

As you can see from these findings, both male and female respondents emphasized the importance of communications and analytical skills. Perhaps the greatest differences occurred in the assessment of the importance of teamwork and organizing. A larger percentage of males than females considered these skills critical to their success.

Critical Skills gained from Education in MIS

In response to the question regarding skills and capabilities gained from educational preparation in MIS which were most critical to successful job performance, the respondents noted systems analysis and programming skills as most important. See Table 3:

Table 3: Critical Skills gained from Education in MIS

	Males	Females	Total
Systems Analysis	.48	.78	.56
Programming	.40	.56	.44
Communications	.24	.44	.29
Database Mgmt	.32	.11	.26
Program Mgmt	.32	.11	.20

As you can see from these findings, a greater percentage of women mentioned the importance of systems analysis and communications skills.

Interview Findings: The Career Experiences of Women

Of the five women who were interviewed, four had achieved promotions to senior systems analyst or project leader in major corporations. In all four cases, they reported receiving career support from a sponsor who served as a supervisor and mentor. In addition, these four women reported having a considerable amount of job autonomy and latitude in accomplishing tasks.

Although they felt "accepted," most of the women interviewed pointed to the "old boy network" within their corporations and thought that they might not get "picked for the team" when special projects and assignments occurred. In one case, a woman witnessed outright sexism and harassment. Even though she viewed this as a byproduct of the "company mentality," she was quite unhappy in her job and shunned the "social side" of the workplace. The only compensating factor, she felt, was that many of the men who had been there much longer than she respected her competence and consulted with her whenever they had questions on technical issues.

With respect to the issue of "met expectations," four of the five women were confident that they had achieved relevant career goals by mastering technical challenges. One had become an EDI (electronic data interchange) expert, while several others had mastered 4GL's (fourth generation languages). While they derived a considerable amount of career satisfaction from these technical challenges, and had received promotions based upon technical merit, most of these women did not have a clear view of the next step in their career paths.

Part of the "problem" these women were experiencing with career uncertainty may be related to the MIS profession itself. As already noted, many MIS jobs rely upon technical skills. Once these technical skills are achieved, an individual can reach a career plateau, with the only way out being a lateral step into a business unit. The other complicating factor was corporate downsizing, which was eliminating a layer of middle-level management.

Interview Findings: The Career Experiences of Men

One of the interesting findings from these interviews was the similarily between the career experiences of the men and women. The men who were interviewed had a variety of careers in MIS: as project leaders (two), a consultant, and an internal auditor. One had left the MIS field altogether and was now pursuing a career track in the mortgage operations end of the insurance business.

One of the interesting aspects of the men's career progression was their mobility. One individual had made a transition from a programming position in a large MIS shop to an end-user consulting position. The end-user consulting position provided him with the opportunity to make a lateral move into the business side as a "systems" person; and once there, he was able to find a sponsor who was willing to teach him the "business" side. In another case, an individual landed a position as a "re-engineering" consultant with a Big-6 accounting firm after a series of positions as a systems specialist--both within MIS groups and within business units.

In contrast to the two men who had made numerous job moves, the other three reported career stability. One had risen to project manager after almost ten years of programming; another was an internal auditor; and another was "plateaued" as a programmer/analyst at a government agency. Interestingly enough, the latter individual felt that his organization was deliberately promoting women and minorities, and "leaving white males behind."

On the issue of "feelings of acceptance," only one of the men felt part of an "old boy network." Three felt that men and women were on a level "playing field" in the MIS profession, because career success at the lower levels depends primarily upon technical knowledge and intelligence, not "informal networking." Women were clearly part of the "team," one project manager noted.

In terms of job discretion, the men who were interviewed reported having considerable job autonomy. This was particularly true of the jobs in end-user computing, project management, and consulting. However, the technical plateau was still a reality for the men, as it was for the women. Three of the five men were not making a transition into management, and in two cases this was intentional. In each case, they had made career progress based upon technical expertise and had reached a plateau.

CONCLUSION

The study of the career experiences of men and women in the information systems field provides some interesting insights. Both men and women felt acceptance and reported having opportunities for job discretion, problem-solving, and decision-making in their positions. Sponsorship and career support were viewed as positive influences in their career development.

Both the men and the women were making progress because of their ability to master the technical challenges, but they were "uncertain" about their opportunities for upward mobility, both because of reaching a technical plateau and because of limited management opportunities in the field. Some were primarily interested in a technical career and did not aspire to management. For others, making a transition into providing information systems support within a functional line of business was a strategy which could provide greater career mobility. It was clear that key skills in information systems careers included effective communications, business understanding, and teamwork.

REFERENCES

Alutto, J.A., Hrebiniak, L.G., and Alonso, R.C.. "On Operationalizing the Concept of Commitment," Social Forces, V. 51, No. 1, June 1973, pp. 448-454.

Baroudi, J.J. "The Impact of Role Variables on IS Personnel Word Attitudes and Intentions," MIS Quarterly, V. 9, No. 4, December 1985, pp. 341-356.

Bartol, K.M. "Turnover Among DP Personnel: A Causal Analysis," Communications of the ACM, V. 26, No. 10, October 1983, pp. 807-811.

Bartol, K.M. and Martin, D.C. "Managing Information Systems Personnel: A Review of the Literature and Managerial Implications," MIS Quarterly, Special Issue, 1982, pp. 49-70.

Berry, Patricia. "Mentors for Women Managers: Fast-track to Corporate Success," Supervisory Management, V. 28, No. 8, August, 1983, pp. 36-40.

Bova, Breda M. and Phillips, Rebecca R.. The Mentor Relationship: A Study of Mentors and Protoges in Business and Academia, ERIC 208 233, 1981, 14 pp.

Burke, Ronald. "The Role of Mentors and Sponsors in Management Development," CTM: The Human Element, V. 15, No. 2, December, 1982, pp. 10-13.

Combs, Jeanne M. and Tolbert, E.L.. "Vocational Role Models of College Women," Journal of the National Association for Women Deans, Administrators, and Counselors, V. 43, No. 1, 1980, pp. 33-38.

Couger, J.D. "Motivators vs. Demotivators in the IS Environment," Journal of Systems Management, V. 39, No. 6, June 1988, pp. 36-41.

Couger, J.D. and Zawacki, R.A. Motivating and Managing Computer Personnel, Wiley, New York, NY, 1980.

Fernandez, J.P. Racism and Sexism in Corporate Life: Changing Values in American Business. Lexington Books, Lexington, MA, 1981.

Ferratt, T.W. and Short, L.E. "Are Information Systems People Different? An Investigation of How They Are and Should Be Managed," MIS Quarterly, V. 12, No. 3, September 1988, pp. 427-443.

Fitt, L.W. and Newton, D.A. "When the Mentor is a Man and the Protogee a Woman," Harvard Business Review, V. 59, No. 2, March-April 1981, pp. 56-60.

Gill, Patricia and Butler, Janet. "Asleep at the Switch," Chief Information Officer Journal, March-April 1993, pp. 34-39.

Greenhaus, J.H., Seidel, C., and Marinis, M. "The Impact of Expectations and Values on Job Attitudes," Organizational Behavior and Human Performance, V. 31, No. 3, June 1983, pp. 394-417.

Greenhaus, J.H., Parasuraman, S., and Wormley, W.M. "Race, Organizational Experiences and Career Outcomes," Academy of Management Journal, V. 33, No. 1, March 1990, pp. 64-86.

Hackman, J.R. and Oldham, G.R. "Motivation Through the Design of Work: Test of a Theory," Organizational Behavior and Human Performance, V. 16, No. 2, April 1976, pp. 250-279.

Harlan, A. and Weiss, C.L.. "Sex Differences in Factors Affecting Managerial Career Advancement," In P.A. Wallace (Ed.), Women in the Workplace, Boston: Auburn House, 1982, pp. 59-100.

Helgesen, Sally. The Female Advantage, New York: Doubleday, 1990.

Hennig, Margaret and Jardim, Anne. The Managerial Woman, New York: Simon and Schuster, 1977.

Ilgen, D.R. and Youtz, M.A. "Factors Affecting the Evaluation and Development of Minorities in Organizations." In K. Rowland and G. Ferris (Eds.) Research in Personnel and Human Resource Management: A Research Annual. Greenwich, Conn.: JAI Press, 1986, pp. 307-337.

Igbaria, M. "Job Performance of MIS Professionals: An Examination of the Antecedents and Consequences," Journal of Engineering and Technology Management, V. 8, No. 2, August 1991, pp. 141-171.

Igbaria, Magid and Wormley, Wayne. "Organizational Experiences and Career Success of MIS Professionals and Managers: An Examination of Race Differences," MIS Quarterly, December 1992, pp. 507-529.

Kanter, Rosabeth Moss. Men and Women in the Corporation, New York: BasicBooks, 1977.

Kanter, R.M. "Differential Access to Opportunity and Power, " in Discrimination in Organizations. R. Alvarez, (ed.), Jossey-Bass. San Francisco, CA, 1979, pp. 52-68.

Kirkley, J. "MIS Profession is Not What it Used to Be," Computerworld, March 21, 1988, pp. 80-81.

Levitin, T., Quinn, R.P., and Staines, G.I. "Sex Discrimination against the American Working Woman," American Behavioral Scientist, V. 15, 1971, pp. 238-254.

McMullen, John. "Panel discusses 'Women and Computers' (Scholarship on Women and Society Conference)," Newsbytes, June 8, 1992.

Morrison, White and Van Velsor. Breaking the Glass Ceiling: Can Women Reach the Top of America's Largest Corporations? Addison Wesley, 1987.

Nixon, R. Black Managers in Corporate America: Alienation or Integration, National Urban League, Washington, D.C. 1985a.

Nixon, R. Perceptions of Job Power Among Black Managers in Corporate America, National Urban League, Washington, D.C. 1985b.

Olsen, C.A. and Becker, B.E. "Sex Discrimination in the Promotion Process," Industrial and Labor Relations Review, V. 36, 1983, pp. 624-641.

Rhode, Deborah. Justice and Gender. Cambridge, MA: Harvard University Press, 1989.

Rosener, Judy B., "Ways Women Lead," Harvard Business Review, November-December 1990.

Schwartz, Felice N.. "Management Women and the New Facts of Life," Harvard Business Review, January-February 1989.

Sumner, Mary, et.al.. "An Assessment of the Attitudes of Graduates and Employers toward Competencies Needed for Entry-Level MIS Positions," Proceedings of ISECON 1990, Chicago, 1990, pp. 129-134.

Tanniru, M.R. "An Investigation of the Career Paths of the EDP Professionals," Proceedings of the 20th Annual Computer Personnel Research Conference, Charlottesville, VA, November 17-18, 1983, pp. 87-101.

Terberg, J.R. and Ilgen, D.R. "A Theoretical Approach to Sex Discrimination in Traditional Masculine Occupations," Organizational Behavior and Human Performance, V. 13, 1975, pp. 352-376.

Turner, J.A. and Baroudi, J.J. "The Management of Information Systems Occupations: A Research Agenda," Computer Personnel, V. 10, No. 4, 1984, pp. 61-68.

Woodruff, Charles K. "Data Processing People: Are They Satisfied/Dissatisfied with Their Jobs?" Information and Management, V. 3, 1980, pp. 219-225.

Women, Computing and C

Douglas A. Newlands & G. Joy Teague
School of Computing and Mathematics
Deakin University
Victoria 3217
Australia
doug@deakin.edu.au, joyt@deakin.edu.au

ABSTRACT

The first language to which students are exposed is most important since it is through that language that they are introduced to the basic concepts and distinctions of computing. Our IS and CS students start with a common course which now uses C but previously used Pascal. Since C is recognized as being a more difficult language than Pascal, it might be expected that changing the introductory programming language from C to Pascal would disadvantage students with little prior programming experience. As women entering a computing courses typically have less prior experience than men it could be hypothesized that the introduction of C as the first programming language in a computing course might prove particularly difficult for the women in the course. To investigate this hypothesis, students' results in the introductory programming course over a three year period were analyzed. Pascal was taught in the first of the three years, and C was taught in the latter two years. Responses to a questionnaire completed by students in the most recent year of the course were examined both in relation to the language issue and to other factors which might precipitate gender differences in performance. This analysis provided evidence that the hypothesis was clearly incorrect although it uncovered an anomaly in the performance of female students in the past.

INTRODUCTION

The choice of the first programming language in a degree course has been cause for discussion since the introduction of computing courses. For many years Pascal was the preferred teaching language and its popularity can be attributed to its ease of use, and to the fact that it was designed specifically as a teaching tool to introduce programming concepts to students (Jensen, 1975). Because of the inbuilt limitations of the language, it never gained wide acceptance in industry, but has been used, almost exclusively, as an introductory programming language.

Over recent years there has been a gradual move away from Pascal and the prime contenders when choosing a replacement language have been C and C++ (Morton and Norgaard, 1993). Ada, Modula-2 and functional languages such as Miranda and Scheme (Hagan, 1992) have attracted a small following and Bernstein has suggested that software packages can be used to provide an introduction to computing concepts which is superior to using procedural languages (Bernstein, 1992). C is preferred by many because of its wide acceptance in industry, because it is the language of choice for PC development work both commercially and in a home environment and, lastly, because it is needed in many subsequent computer science topics (Newlands 1992). C can be described as a small language with a rich array of optional libraries, permitting low level access to hardware but also providing high-level constructs for applications programming (Kernighan 1978, Hutchison 1988, Gottfried 1990). The opponents of C have pointed out some problems with it (Mody 1991) but, apart from one particular lexical problem (equality test and assignment operator), in our experience they have little relevance for students learning their first programming language.

Our experience has been that, if sufficient support is given to students, the transition from Pascal to C has little effect on students' results (Newlands 1992a). The other innovations which were introduced with the change to C were (i) to spend the first five weeks of the semester teaching algorithm construction, using Structured English, prior to the introduction of the programming language and (ii) to have students complete programming exercises, which reinforced the lecture material, in formal laboratories for the first half of the semester and to use traditional assignments in the second half. When Pascal was taught, traditional assignments, completed individually by the students in their own time, were used for continuous assessment and there were no scheduled laboratory classes. Interviews with students at the end of the first C course confirmed that laboratory exercises were preferred by a majority of students over assignments done in their own time. Consequently, in the second year in which C was taught, the formal laboratory exercises were continued throughout the semester and traditional assignments were abandoned. These exercises contributed 40% of the students' assessment. It is commonly recognized that (i) laboratories are an essential part of a computing course (Denning, 1988) and (ii) women are more likely to complete their course if they are learning in 'structured labs using group projects based upon cooperative learning models' (Martin and Murchie-Beyma, 1992, p. 7).

The literature suggests that, on average, men entering a computing course will have a higher level of computing experience than the women entering a course. A greater proportion of boys than girls study computer science subjects at secondary school (Taylor, 1992). Clarke lists many sex differences in experience. For example, boys are more likely to have access to a computer at home, receive more encouragement from fathers, are more likely to participate in summer school computing camps and are the major patrons of video games arcades (Clarke, 1992). Frequently the features of the computing environment in which boys operate alienate girls (Clarke, 1991). There is a general belief, perhaps as a result of the above, that girls are not good at computing (Clarke, 1992).It might, therefore, be expected that a change from the more straightforward Pascal, to C, claimed by many as a user unfriendly language (e.g. Mody, 1991) would disadvantage students with little previous computing experience. The literature indicates that the majority of these students are likely to be women (Howell, 1993).

MOTIVATION FOR THE STUDY

Concern that the introduction of C might reduce the already sparse sprinkling of women in computing classes prompted an analysis of student results over three semesters of teaching the introductory programming unit. In addition, a questionnaire was given to students at the end of the most recent one semester course. The course is offered only in the fall semester and has been taught by the same person for each of the three years, one year of Pascal, then two years of C. Student evaluations and formal interviews with students by another staff member had confirmed that the teacher was considered by the students to be an excellent teacher. The analyses of each factor which might lead to a gender-related difference in performance are presented below.

FINDINGS

Effect of language on course choice and completion

The number of students and percentages of men and women completing the course are given in Table 1 (students who withdraw are omitted since most of these withdraw before the start of lectures and the majority of the remainder do so before the first practical work is due, and their inclusion does not alter the outcome).

TABLE 1: Gender Distribution of Intake

	Males		Females	
Year	No.	%	No.	%
1991	114	75	37	25
1992	104	71	43	29
1993	128	73	47	27

As can be seen from the table, the ratios have remained consistent over the past three years showing that the number of women choosing computing has not been affected adversely by the change from Pascal to C. Testing the hypotheses that "year A male/female ratios are not different from year B ratios" for all combinations yielded the following results:-

Year	Chi-square	Critical value	Significance
91/92	0.855	3.84	0.05
92/93	0.227	3.84	0.05
91/93	0.235	3.84	0.05

In no case is there evidence that the hypothesis is untrue so the introduction of C has had no effect on the attractiveness of the course to either gender group. As many students have little previous programming experience and are probably unable to make informed judgments of the relative merits of C and Pascal these results may be misleading.

Effect of language on grade distribution within gender

If there is a difference in male/female performance in computing subjects, it is to be expected that it will be reflected in the difference in the distribution of grades between the sexes. The distributions, for the last three years, are shown in Table 3 (the correspondence between grades and raw marks is shown in Table 2.

Table 2. Grade-Mark Equivalence

Grade	HD	D	C	P	N
Mark	80-100	70-79	60-69	50-59	0-49

Table 3. Distribution of Grades by Gender

	1991		1992		1993	
No. enrolled	M=114	F=37	M=104	F=43	M=128	F=47
Grade	%	%	%	%	%	%
HD	12	3	8	7	7	9
D	18	19	17	16	10	13
C	18	41	21	26	24	23
P	25	27	27	26	19	19
N	26	11	27	26	40	36

If we examine the hypothesis that "there is no difference between the performance of male and female students" for each year we find that:-

Year	Chi-square	Critical value	Significance
1991 (Pascal)	11.96	9.48	0.95
1992 (C)	0.347	9.48	0.05
1993 (C)	0.449	9.48	0.05

For 1991, the hypothesis can clearly be rejected but not for the other years. This seems to provide firm evidence that women are not disadvantaged by teaching C. However it uncovers the unexpected fact that, in the year in which Pascal was taught, the distribution of women's results was very different from the men's results. To examine this more closely, the proportions of male and female students obtaining the higher grades were examined in Table 4.

Table 4. Distribution of Higher Grades by Gender

	1991		1992		1993	
No. enrolled	M=114	F=37	M=104	F=43	M=128	F=47
Grade	%	%	%	%	%	%
HD/D	31	22	25	23	17	21
Other	69	78	75	77	83	79

Testing the hypothesis that "there is no difference between male and female performances", the chi-squared values are 1.13, 0.05, 0.35 for 1991, 1992 and 1993 respectively. Since the critical value is 3.84 at significance level 0.05, there is no reason to reject the hypothesis and we can conclude that there are no gender differences in performance in the higher grades.

When the performance of women at the pass/fail interface is examined (see Table 5), it is found that there is a significant difference only for 1991(chi-square = 3.849, critical value 3.841 at significance 0.05; for 1992 and 1993 the chi-square values are 0.25 and 0.195 respectively).

Table 5. Distribution of Lower Grades by Gender

	1991		1992		1993	
No. enrolled	M=114	F=37	M=104	F=43	M=128	F=47
Grade	%	%	%	%	%	%
Pass	74	89	73	74	60	64
Fail	26	11	27	25	40	36

Thus it can be seen that the female failure rate was significantly lower than the male rate when Pascal was taught. This difference does not occur in later data and it is clear that neither sex is advantaged presently. The historical better performance of the women is surprising given that they enter the course with less experience but, as data are limited, reasons for it must remain speculative.

Effect of examination style on student performance

In 1993 multiple choice questions were introduced in the end of semester examination and they constituted about half of the final examination. It has been suggested that, at tertiary level, women may be slightly disadvantaged by this form of assessment (Hegarty-Hazel et al., 1993). An analysis of the multiple choice section of the examination revealed that both sexes had performed very similarly with identical mean marks and standard deviations which were not significantly different (F test at 0.05). So the multiple-choice examination has not had any adverse effect unless there are reasons to suspect that the female students should have done better than the male students. If there were any such objective reason for this belief it might reasonably be on the basis of school-leaving results of the students. The VCE (Victorian Certificate of Education - secondary school certificate; basically the sum of best 4 subjects including English; converted to a percentage here for convenience) scores of both sex groups are shown in Table 6 along with the group performance in the first computing course.

Table 6. Secondary and Tertiary Results

	VCE		BPC	
	Mean	St. Dev.	Mean	St. Dev.
Male	62.48	7.53	55.82	12.23
Female	63.27	9.39	52.6	13.64

It is noticeable that the female VCE score is higher than the male VCE score but that the female score in Basic Programming Concepts (BPC) is lower than the male score. Statistically, these scores are not different at a significance of 0.05. Therefore, these data provide no evidence that the women performed differently from the men in the VCE. Thus there is no reason to expect that the women would have performed better than the men in the multiple choice examination. Therefore, it can be concluded that the multiple choice questions did not significantly disadvantage female students

One surprising fact did come to light when we examined the correlation between students' VCE and BPC scores. For male students the correlation was 0.482 which seems low but for female students it was just 0.336. This would suggest that VCE scores are not a good predictor for a female student's success in a computer programming course (at least for students in the same percentile range of the state school leavers as we accept).

Effect of Previous Computing Experience

From the questionnaire we can assess other aspects of male/female preparedness for introductory computing. Students were asked about their previous computing experience and the results are shown in Table 7.

Table 7. Previous computing experience

	Women		Men	
Skills	No.	%	No.	%
None	22	61	27	46
Limited	6	17	14	24
Extensive	8	22	18	30

In this study, a larger proportion of women (61%) than men (46%) claimed no previous computing experience and only slightly more men (30%) than women (22%) claimed extensive experience. A similar earlier study of students majoring in computing found that almost half of the women interviewed (48%) had no (or minimal) computing experience prior to entry to the course, while only 17% of men had no (or minimal) previous experience (Teague & Clarke, 1991). In both cases, the women are clearly less well-prepared for the course. The larger numbers of male students found in this study with little preparation is, perhaps, a consequence of there being a large increase in the number of students taking the course as a service course in 1993.

As questionnaires for the current study were anonymous, it was not possible to determine the relationship between previous experience and intention to major in computing, however a comparison of the two sets of data suggests that the influx of service students may have biased the computing skills data.

Programming skills

Students were asked to evaluate their programming skills on entry to university (Table 8). Again, more men (51%) than women (37%) indicated previous experience, but again most of this experience was limited to introductory programming at home or school. These data also suggest that the men are only marginally advantaged by their previous programming experience when compared with the women in the course (there is no statistically significant difference in programming experience between the males and females: chi-square = 2.9 with critical value 7.8 at 0.05), although men initially tend to be more optimistic about their chances of success in such a course(Clarke and Chambers, 1989).

Table 8. Programming skills

	Women		Men	
Skills	No.	%	No.	%
None	22	63	29	49
Average	11	31	21	36
Superior	2	3	7	12
V. Superior	0	0	2	3

A detailed analysis of data from the earlier study (Teague & Clarke, 1991) suggests that slightly more than one third of female computing majors (39%) interviewed for that study had previous programming experience, compared with approximately three quarters of male students (77%). As before, this suggests that the increased numbers of service students in 1993 may be skewing the data.

CONCLUSIONS

The concern that teaching C as the first language in a computing course may be disadvantaging the women entering the course appears to be unfounded. Our evidence suggests that they are achieving results comparable to men in the first programming course, despite all the factors which one might expect to disadvantage them. Indeed our analysis suggests that most of these factors are no longer operating in our environment. Although women are shown to still be less well prepared than male students at entry to the course, there is no evidence that there is any change in the gender mix of the class as a result of the language change. Nor did we find any difference in the performance of the gender groups while using C although, in the Pascal course, it seemed that women were less likely to fail than men. Similarly, we could find no difference in gender group performance as a result of the introduction of multiple choice testing either in terms of actual examination performance or in terms of expectations from secondary school performance.

We would hypothesize that C is not so very different from Pascal, that the traditional lecture/assignment regime has made introductory programming difficult for new students of both genders and that the new laboratory-oriented teaching has broken this cycle. We believe that both female and male students can benefit from improved teaching methods and support structures in computing courses.

REFERENCES

Bernstein, D. (1990). A new introduction to computer science. In C. D. Martin & E. Murchie-Beyma (Eds.) *In Search of Gender-Free Paradigms for Computer Science Education*, National Educational Computing Conference, Nashville, Tennessee, 25- 27 June, 87-91.

Clarke, V. A. (1991). Creation of female-friendly computing environments. In L. J. Rennie, L. H. Parker & G. M. Hildebrand (Eds.) *Action for Equity: The Second Decade. Contributions to the Sixth International GASAT Conference*, 1, The University of Melbourne, Victoria, Australia, 14-19 July, 27-35.

Clarke, V. A. (1990). A new introduction to computer science. In C. D. Martin & E. Murchie-Beyma (Eds.) *In Search of Gender-Free Paradigms for Computer Science Education*, National Educational Computing Conference, Nashville, Tennessee, 25- 27 June, 71-86.

Clarke, V. A. & Chambers, S. M. (1989). Gender-based factors in computing enrollments and achievement: evidence from s study of tertiary students. Journal *of Educational Computing Research*, 5(4), 409-429.

Gottfried, B. S. (1990). *Programming with C*, McGraw Hill.

Hagan, R. (1992). Teaching multi-paradigm programming. *Proceedings of the 1992 Conference of the Australian Society for Computers in Learning in Tertiary Education*, 307-315.

Hegarty-Hazel, E., Logan, P., & Gallagher, P. (1993). Gender and assessment of students in physics. In S. Haggerty & A. Holmes (Eds.) *Transforming Science and Technology: Our Future Depends On It. Contributions to the Seventh International Gender and Science and Technology Conference*, 2, The University of Waterloo, Ontario, Canada, 31 July - 5 August, 773-781.

Howell, K. (1993). The experience of women in undergraduate computer science: what does the research say?. *SIGCSE Bulletin*, 25(2), 1-8.

Hutchison, R. C. & Just, S. B. (1988). *Programming Using the C Language. McGraw*-Hill.

Jensen, K. & Wirth, N. (1975). *Pascal User Manual and Report. Springer*-Verlag, 133.

Kernighan, B. W. & Ritchie, D. M. (1978). *The C Programming Language. Prentice*-Hall.

Martin, C. D. & Murchie-Beyma, E.(1992). *In Search of Gender Free Paradigms for Computer Science Education*, Publication

Mody, R. P. (1991). C in education and software engineering. *SIGCSE Bulletin*, 23(3), 45-56.

Morton, L. & Norgaard, N. (1993). A survey of programming languages in CS programs. *SIGCSE Bulletin*, 25(2), 9-11, 18.

Newlands, D. A. (1992). C as a First Programming Language. *Proceedings of the 1992 Conference of the Australian Society for Computers in Learning in Tertiary Education*, 339-345.

Newlands, D. A. (1992a). *Teaching Programming to First Year Students*. Deakin University Technical Report C92/17.

Taylor, C. (1992). Contained in a report presented to the Victorian Curriculum and Assessment Board, Information Technology Field of Study.

Teague, G. J., & Clarke, V. A. (1991). Fiction and fact: Students' and professionals' perceptions of women in computer science. In I. V. Ericksson, B. A. Kitchenham, & K. G. Tijdens (Eds.), *Women, Work and Computerization: Understanding and Overcoming Bias in Work and Education* (pp. 363-375). Amsterdam: North-Holland.

Willoughby, L. & Carter, R. (1993). Electronics education for women: evolving new approaches. In S. Haggerty & A. Holmes (Eds.) *Transforming Science and Technology: Our Future Depends On It. Contributions to the Seventh International Gender and Science and Technology Conference*, 2, The University of Waterloo, Ontario, Canada, 31 July - 5 August, 865-873.

INTEGRATING GRAPHICS INTO DATA STRUCTURES AND ALGORITHMS IN THE DPMA MODEL CURRICULUM

Alfred L. McKinney
Krishna K. Agarwal
Reza Sanati

Department of Computer Science
Louisiana State University in Shreveport
One University Place
Shreveport, LA 71115

ABSTRACT

This paper recommends the introduction of an integrated sequence in data structures, algorithms and graphics into the standard Information Systems DPMA Curriculum for a Four Year Undergraduate Degree for the 1990's. The rationale is that this integrated sequence will provide a better understanding of these fundamental and central areas of information systems through immediate application and visualization. It will also provide an exposure to graphics and Graphics User Interfaces (GUIs), and an opportunity to develop a significant graphics package to aid in the animation and visualization of the data structures and algorithms.

INTRODUCTION

It is well established by consultant reports, model curriculum guides, and expert opinion that data structures and algorithms are the very heart and core of Information Systems (IS). This means that many information systems programs require courses in data structures and algorithms.

Many consider graphics to be important and offer a graphics course as an elective. Computer Graphics (CIS 10) was an elective course in the DPMA 1986 curriculum and is listed as an elective in the IS '90 curriculum [DPMA 1991]. We are proposing that graphics be integrated into courses dealing with data structures and algorithms. This integration of graphics is further supported in the following paragraph.

We are now in the third generation of computer applications. This generation is dominated by computing with graphical objects. This change in emphasis was triggered by the advent of computers with bitmap graphics. This has led to the widespread use of sophisticated Graphical User Interfaces (GUIs) and a great increase in applications such as Computer-Aided Design (CAD), image processing and pattern recognition (in medicine, cartography and robotics) [Nievergelt 1993]. Presentation graphics, animation, desk-top publishing graphics and graphics modeling systems are becoming prominent and widespread. Graphics is now a dominant area of information systems and it should be moved from the elective category to the required.

REASONS AND BENEFITS

Graphical User Interfaces (GUIs) have become the major means of interacting with information systems. It has been predicted that the next wave of information systems development will utilize graphics in a windows environment. We need to prepare our students for this by requiring them to study graphics.

Graphics is rich in data structures and algorithms. By including the study of graphics along with that of data structures and algorithms we are able to provide immediate applications, illustrations and use of data structures and algorithms. This leads to a much better understanding of these fundamental areas of information systems. Also it provides a stronger foundation which will better prepare the students for their more advanced information systems courses.

Another advantage of requiring graphics is that it gives an excellent opportunity to take a vertical approach in introducing an object-oriented approach in development. Many information systems programs are taking a vertical approach to implementing or incorporating object-oriented design and programming (the vertical approach is to introduce object-orientation in one or two courses in the program [Waguespack 1993]).

INTEGRATING GRAPHICS INTO THE IS MODEL CURRICULUM

Presently, the IS Model Curriculum requires Data Structures and Algorithms and has Graphics as an elective, [DPMA 1991]. We propose that Graphics be integrated at least into the Data Structures and Algorithms courses.

An illustration of how graphics could influence the teaching of data structures is as follows: the array data structure could be introduced and then demonstrated by using arrays to represent coordinate endpoints of line segments. Then line algorithms such as Digital Differential Analyzer (DDA) and Bresenham could be presented. This should be the approach to teaching the sequence -- introduce a data structure, use it in a graphics representation and illustrate with an associated algorithm. Graphics algorithms are usually computationally intensive so there is ample opportunity to focus on developing efficient algorithms.

The data structures and algorithms used in graphics seem to provide a natural and powerful approach to teaching and illustrating data structures and algorithms. One approach would be to include the development of a simple graphics package -- at least a 2-Dimensional package. This will serve several purposes. It will provide a goal-oriented approach plus an opportunity to immediately illustrate what has been introduced. Also, it provides experience in analyzing and implementing algorithms. Further, the development of this package will require utilizing and illustrating data structures such as arrays, linked lists and quadtrees.

The graphics package can be developed throughout the teaching of data structures, algorithms and graphics. The graphics package should be menu driven and should provide the capability to draw output primitives such as lines, circles, and ellipses. This will permit an opportunity to analyze and implement the Bresenham line algorithm and the scan-line algorithm. Also, the Cohen-Sutherland Line Clipping Algorithm can be implemented. Others such as area clipping and hidden-line removal seem appropriate to implement.

The transformations of translation, scaling and rotation should be included in the package. This is an excellent opportunity to illustrate the advantage of homogeneous coordinates and of concatenation for composite transformations [Hearn 1994].

If time permits a simple 3-D wireframe model of an object such as the famous Utah Teapot can be implemented along with the 3-D transformations of scaling and rotation. This can be done with Bezier curves [Watt 1993].

If time permits, classroom demonstrations of a presentation system such as Harvard Graphics could be made. Also, it would be appropriate to demonstrate a desktop publishing system such as Ventura.

This added integration of and exposure to graphics will facilitate the use and understanding of a graphics system such as ISDT to illustrate data structures [Lin 1993]. Also, it will be very helpful in the animation of algorithms with a graphics system such as Zeus [Brown 1992] or with the use of specialized programs such as those in [Sigle 1987].

PROPOSED LABORATORY PROJECTS

Some possible computer lab projects for the graphics component of the integrated sequence are:

1. Begin to develop a 2-Dimensional Color Graphics Software Package. Develop the menu and features to set graphics mode, set window, set background, foreground and draw colors, and to draw a line or lines using the Bresenham line algorithm.

2. Extend the 2-D graphics package by adding features to draw circle(s) and ellipse(s). Use the Bresenham algorithm to draw the circle(s) and a modification of the algorithm to draw the ellipse(s).

3. Extend the 2-D graphics package by adding the capabilities: to enter a set of coordinate points (and optionally a fixed point), draw lines connecting current set of points, transform (translate, scale, rotate or a composite transformation), draw lines connecting set of transformed points, and update old set of points with transformed points.

4. Design a graphics program for a business graphics application such as comparison bar charts. The program should be somewhat general in nature. It should provide for the user to enter headings and labels. Also, it should automatically scale the data and display a color legend indicating the comparison.

5. Write a simple animation program with at least two different images and several movements. If there is not enough time for the animation lab project, then each student can be assigned the task of writing a brief paper on animation.

PROPOSED TEXTS AND SYLLABI

Possible texts for the proposed integrated sequence are [Nievergelt 1993] and [Kruse 1987]. A possible graphics reference text as a supplement is [Foley 1994] or [Hearn 1994]. A good graphics programming supplement is [Ezzell 1990] or [Ezzell 1991]. Possible course outlines are given in the appendix.

CONCLUSION

We feel strongly that great gains would be obtained by a required integrated sequence in data structures, algorithms and graphics. These gains are:

- a better understanding of data structures and algorithms through immediate application and visualization,
- an exposure to graphics and GUIs,
- the development of a significant graphics package, the use of a GUI-based tutoring system such as ISDT for learning Data Structures [Lin 1993],
- the use of a graphics system such as Zeus to animate algorithms [Brown 1992].

We further strongly recommend that this paradigm be implemented in future IS curricula.

REFERENCES

[Brown 1992] Marc H. Brown, "Zeus: A System for Algorithm Animation and Multi-View Editing", SRC Research Report 75 by Digital, February 28th, 1992, 23 pages.

[DPMA 1991] DPMA, Information Systems: The DPMA Model Curriculum for a Four Year Undergraduate Degree, Data Processing Management Association, 1991.

[Ezzell 1990] Ben Ezzell, Graphics Programming in Turbo C++, Addison-Wesley Publishing Company, Inc., 1990.

[Ezzell 1991] Ben Ezzell, Graphics Programming in Turbo Pascal 6.0, Addison-Wesley Publishing Company, Inc., 1991.

[Foley 1994] James Foley, Andries van Dam, Steven Feiner, John F. Hughes, and Richard Phillips, Introduction to Computer Graphics, Addison-Wesley Publishing Company, 1994.

[Hearn 1994] Donald Hearn and M. Pauline Baker, Computer Graphics, Prentice-Hall, 1994.

[Kruse 1987] Data Structures and Program Design, 2nd edition, Kruse, Prentice-Hall, 1987.

[Lin 1993] Billy Lin, Jean Wang, Chris Kohlbrecker and Jamie Jason, "IDST: A GUI-Based Tutoring System for learning Data Structures", 1993 ISECON Proceedings, Nov., 1993, pp. 62-69.

[Nievergelt 1993] Jurg Nievergelt and Klaus H. Hinrichs, Algorithms and Data Structures With Applications to Graphics and Geometry, Prentice-Hall Inc., 1993.

[Sigle 1987] John W. Sigle, "Bringing Algorithms to Life", ACM Annual Computer Science Conference Proceedings, February 1987, pp. 327-335.

[Waguespack 1993] Les Waguespack, "Integrating Object Orientation in the IS Curriculum", 1993 ISECON Proceedings, Nov., 1993, pp. 169-176.

[Watt 1993] Alan Watt, 3D Computer Graphics, 2nd edition, Addison Wesley, 1993.

APPENDIX

The following are course outlines for the proposed integrated sequence in data structures, algorithms and graphics:

DATA STRUCTURES, ALGORITHMS AND GRAPHICS I

DESCRIPTION: A study of the primary data structures used in computing, their definitions as abstract data types and some of their possible implementations. The design and analysis of algorithms which are useful in the manipulation of these data structures. Techniques for representation, transformation, and display of patterns and images on display devices.

TEXT: Algorithms and Data Structures With Applications to Graphics, Nievergelt and Heinrichs, Prentice-Hall Inc., 1993.

SUPPLEMENTS: Data Structures and Program Design, 2nd edition, Kruse, Prentice-Hall Inc., 1987.

OBJECTIVES: To gain knowledge of and experience with the basic data structures and algorithms of information systems utilizing a graphics-oriented approach. To gain an appreciation of the concept of abstract data structures and their role in the design of software systems. To introduce the basic principles, concepts, terminology, data structures and algorithms of computer graphics.

OUTLINE:

1. Principles of Program Design
2. Introduction of Computer Graphics
3. Graphics Primitives and Environments
4. Graphics Systems
5. Abstract Data Types
6. Arrays, Records, Stacks, Queues, Lists and their Implementation
7. Linked Lists, Dynamic Memory Allocation and Pointers, Linked Stacks and Queues, Applications of Linked Lists
8. Searching (Sequential and Binary)
9. Basic File Organizations, Secondary Storage Devices, Blocking and Buffering
10. Tables, Hashing
11. Recursion
12. Trees (Binary, Balanced and B-Trees and their Traversals)
13. Interactive Graphics Input Methods

DATA STRUCTURES, ALGORITHMS AND GRAPHICS II

DESCRIPTION: A continuation of Data Structures, Algorithms and Graphics I.

TEXT: Primary text same as for Data Structures, Algorithms, and Graphics I.

SUPPLEMENTS: Computer Graphics, Hearn and Baker, Prentice-Hall Inc., 1994 or Introduction to Computer Graphics, Foley, van Dam, Feiner, Hughes, and Phillips, Addison-Wesley Publishing Company, 1994.

OBJECTIVES: To gain knowledge of data structures and algorithms that are useful for problems that arise in information systems. To gain this knowledge via an integrated graphics-oriented approach.

OUTLINE:

1. Notation and Terminology for Algorithmic Development
2. Sorting (Insertion, Quicksort, Mergesort, Heapsort, Shellsort and External Sorting)
3. Strings
4. Matrices and Graphs
5. Two Dimension Transformations of Graphical Objects
6. Windowing and Clipping
7. Hidden-Line Removal
8. Fractals
9. Three Dimensional Graphics
10. Modeling Methods
11. Design of the User Interface
12. Animation Algorithms

Major Revisions are Overdue in Introductory Business Programming Courses

Larry Brumbaugh
Applied Computer Science
Illinois State University
Normal, IL 61761
(309) 438-8019 brumbaug@ilstu.edu

ABSTRACT

During the past two decades few changes have been made in the manner in which business application programming is taught. Many programming courses focus almost exclusively on batch processing and ignore or trivialize the existence of on-line applications. Business programming courses should be redesigned to make their content more relevant to the contemporary data processing world. Six common arguments for maintaining the batch oriented status quo in business programming courses are disputed.

Introduction

Each year many people take courses whose primary function is to help prepare them for careers as business application programmers. One essential skill that such a course should impart is the ability to write standard or typical programs. In reality, most introductory programming courses today teach the same skills which were required and utilized by application programmers 10 or more years ago. While course content has remained essentially static over the past decade, business data processing has evolved, and a different type of programming along with a new set of accompanying skills is now required. The methods and skills of 10 years ago continue to be used by fewer people. On-line applications have probably replaced batch as the fundamental programming mode (5). During the past decade many schools have made few changes in how introductory programming courses are taught, even when other segments of the data processing and computer science curriculum kept pace with changes. This paper recommends a major revision in business programming course content.

Classifying Programs as Either Batch or On-Line

This paper discusses two broad programming categories called "batch" and "on-line". A batch program runs in it's own region performing a mixture of I/O operations with disk, printer, in-stream, and tape data sets. The table below identifies several standard batch application programs. An on-line (sometimes called interactive) program transmits data to and from a terminal screen, may use a teleprocessing (TP) monitor, and run in the TP monitors' region as a subprogram. Typical on-line applications begin by sending a menu to a terminal screen, a user enters data on the screen and the program receives and processes the data. Thus, the program uses the terminal screen itself as the primary input and output device.

Edit a File of Transaction Records
Sequential Master File Update
Random Master File Update
Control Break Program to Generate Printed Reports

Common Batch Application Programs

In the early days of data processing (1950's through mid-1970's), most individual applications and entire systems were batch. Large numbers of on-line applications began to be developed in the mid-to-late 1970s (4). Table 2 shows how the approximate estimated percentages of new batch and on-line applications have changed during the past 15 years.

Time Frame	% of Batch Applications	%On-line Applications
Mid-1970's	Over 90%	Less than 10%
Today - 1994	Approximately 50%	Approximately 50%
5years from now	Less than 33%	Over 66%

Percentage of New Batch and On-line Applications

Many of the batch application programs ever needed by the typical data processing shop have been developed and are in use. In fact, batch programmers today primarily perform maintenance on existing systems. Because of it's recent origins, at many data processing shops some common on-line applications have yet to be implemented or even discovered.

Work performed by on-line programmers can be broadly classified into three areas: conversion or duplication of a comparable existing batch system, developing an on-line system with no batch equivalent, and maintenance programming on existing on-line systems. Because on-line systems are relatively new and since there are fewer of them at present, there is a smaller percentage of maintenance programming than with batch systems. An on-line programming position often allows a person a greater degree of intellectual creativity than a comparable batch programming Job.

As data processing continues to move toward an even larger percentage of on-line applications, it seems certain that batch programming will never again be the primary type of application. Some batch programming will always be necessary, but on-line programming will account for most of the new systems that will be developed. These are conservative statements concerning the direction in which business programming is moving.

Training Business Application Programmers Today

How realistic are the programming courses currently taught to prepare people for the present day data processing environment? This is still the world of (primarily) IBM, DEC and other major computer vendors whose systems are used for business operations. Many university DP/MIS/CS departments appear to ignore the state of business data processing. Since it is apparent that COBOL will remain the overwhelming language of choice for business applications during the foreseeable future, most business programming courses use COBOL. However, many business programming courses teach only batch COBOL programming. If on-line programming is even mentioned, it is usually as a minor topic late in the course.

Batch Verses On-Line Programming Courses

There are several significant reasons for the gap between the batch programming courses currently taught and the environment found in many data processing shops. First, batch programming developed initially and has a much longer history. Second is the belief that a knowledge of batch programming is a prerequisite for learning on-line programming. Third, many people believe that on-line programming is much more difficult. Fourth, batch programming ordinarily uses structured code, while much on-line programming is unstructured. Fifth, some people believe that batch and on-line programming are disjoint entities which should be considered separately. Finally, some people believe that on-line applications can at present only be taught on a microcomputer, and most present day business applications are still done on a mainframe. The next six sections attempt to dispel the above opinions.

The Historic Bias Favoring Batch Programming

On-line programming requires learning some skills not used in a batch processing environment. Conversely, some batch programming concepts are irrelevant in an on-line environment. Some people who develop and teach programming courses are slow to make changes which will obsolete knowledge and skills that took a long time to develop. Such people have the philosophy that if it was good enough for many previous years, took a considerable effort to learn and it still works today, why replace it with something new? There currently are some programming courses taught where the material covered is about a decade out of date.

Is Batch Programming a Prerequisite for On-line Programming

Must a person have batch programming experience before becoming an on-line programmer? When the historic bias that batch programming has always been learned first is discarded, little remains to support the batch programming before on-line programming school of thinking. Initial batch programming projects usually involve reading in-stream data and producing a printed report. Following this, disk data sets are used for input, output, or both. With on-line programming a terminal screen is used as both the primary input and output device, and disk data sets are used for input and output. There is a strong syntactic and functional similarity between I/O operations with a disk and a terminal. A terminal can be thought of as an I/O device which holds exactly one record. Within most programming environments a syntax like that shown in the Table below is used to process disk and terminal data. The parameters in parenthesis are used when processing within a program's actual region in memory (WORKING-STORAGE in COBOL). If they are omitted, processing is performed in a buffer.

I/O Access Type	Disk I/O Syntax	Comparable Terminal I/O Syntax
Input	READ Record(Into)	Receive Screen Contents (Into)
Output	WRITE Record(From)	Send Screen Contents (From)
Input and Output	READ and REWRITE	Converse (INTO...FROM)

Comparable Syntax for Disk and Terminal I/O

Although there are many similarities between disk and terminal I/O operations, on-line programming requires some knowledge concerning the way data on a screen is stored and referenced. Batch programming spends less time examining the structure and function of a disk. In conclusion, it should not be appreciably more difficult to teach entry level programmers to create simple on-line applications than to write batch applications. It can be argued that there are more similarities than differences between the two types of programming. On-line programming concepts are taught to beginning programmers in languages other than COBOL. Batch programming methods can be covered later as an alternative approach.

On-line Programming is More Difficult than Batch

Many people believe that an on-line application is much more difficult to develop than a comparable batch application. If a person with years of batch programming experience and limited on-line programming experience works on comparable programs of each type, they will probably conclude that on-line programming is more difficult. At present, there are very few entry level programmers with an equal mixture of batch and on-line experience. People who have programmed for a number of years exclusively in an on-line environment may find batch programming unusual and confusing when they initially return to it.

Specific reasons given for thinking on-line programming is more difficult than batch include pseudo or non-conversational programming techniques, data integrity problems, saving data between tasks, reducing response time, and processing screen fields. However, if a beginning programmer is taught to write non-conversational programs, then batch or conversational programming may seem difficult or unnatural. Likewise, as hardware and system software have improved, methods of safeguarding data and reducing response time have become less a programmer's problem. Finally, terminal screen programming skills can be learned. In conclusion, there are no outstanding difficulties that prohibit a programmer from beginning with on-line applications.

On-line Programming is Unstructured

Batch application programming developed in the 1950s and early 1960s. Approximately a decade later, modern structured programming evolved (3). The structured concepts were directly applicable to batch programming. On the other hand, ad hoc programming standards developed for on-line application programming. Much of the reason for this is sloppiness and lack of discipline on the part of the programmers and the data processing shops. There is nothing about on-line programming which prohibits it from being structured. CICS is the single most common on-line business programming environment (4). Much CICS code has a well deserved reputation for being unstructured. However, there is nothing which prohibits structured CICS programs (6).

Are Batch and On-line Application Programs FunctionallY Different?

Many standard business data processing applications performed with batch and on-line programs are functionally similar. However, the manner in which the result is achieved is often different. The Table below lists some of the most common batch business data processing applications and identifies an on-line equivalent whenever one exists. Conversely, some important on-line applications listed here do not have a comparable batch equivalent. In conclusion, there are substantial similarities between batch and on-line applications.

Standard Batch Business Application Program	Comparable On-Line Implementation of the Application - When One Exists
Editing transaction records	Edit one record per screen per task
Submit edited records to an update program	Data collection system to store edited transaction records on disk
Random master file update	One record per screen; May require multiple tasks; and necessary to confirm delete and update operations
Sequential master file update	No equivalent or same as random above
Control break program and Report generator	On-line listing with browsing and scrolling
Fancy printed report	Special case of previous entry
Sorting application	No common equivalent
External subroutine call	Transferring to a different transaction
Processing records in reverse	Scrolling backwards
No common equivalent	Selecting one record from a listing for more detailed information
No common equivalent without manual intervention	Automatically initiate processing at a specific (relative or absolute) time

Comparable Batch and On-line Applications

What are Realistic On-Line Implementation Environments?

The final objection against teaching on-line concepts in an introductory business application programming course is that there is no environment where this can be meaningfully done. This is not true. There are many ways in which COBOL programming courses can meaningfully develop on-line applications. It is merely a matter of selecting one or more environments which meet as many other course goals as possible. Available approaches include COBOL programming on a microcomputer, COBOL programming on a minicomputer, mainframe COBOL programming using a TP-monitor such as CICS or IMS, and batch COBOL with subroutine calls to low level (assembler) access method subroutines which perform the terminal I/O. (2) All of these methods can be used to teach on-line programming concepts to relatively inexperienced COBOL programmers. (1)

These methods and others will become much more widespread and user-friendly in the foreseeable future. Hence, an even larger group of possibilities will be available for teaching on-line programming in the microcomputer and mainframe environments.

Conclusion

This paper claims that beginning programmers can as easily be initially taught on-line programming concepts rather than strictly batch programming techniques. Since most future applications will not involve batch programming, a transition in teaching philosophy should be strongly considered. However, at present there is little indication that this transition has begun. This can be confirmed by examining the contents of existing business programming courses and the substantial majority of COBOL textbooks.

At present there are few good textbooks which contain a significant amount of material describing on-line programming from the viewpoint of a beginning or entry-level application programmer. This needs to be remedied immediately to implement the ideas proposed here.

Significant differences also exist between batch and on-line systems with regard to program debugging and testing. These two topics require modifications in any curriculum revision. On-line debugging software and testing techniques have already become standard tools of the practicing application programmer. A COBOL programming course usually does not discuss screen generators, interactive debugging tools, and testing techniques.

Thus, the overall conclusion is to replace the emphasis on batch programming and other batch related topics with their on-line equivalents or at least provide an equal mixture of the two. There is no reason for teaching outdated techniques and concepts merely because it is historic precedent. At present the first step in becoming a business application programmer is to learn to write batch COBOL programs. It will be just as easy and more relevant if the word "batch" is replaced by "on-line and batch".

Eventually the realization will occur that the ideas proposed here are the direction in which the teaching of application programming must move. It is then necessary to decide the best way to make the transition. The transition from teaching batch programming to teaching on-line programming appears to be inevitable.

References

(1) Brown, Gary D, Advanced ANSI COBOL With Structured Programming, John Wiley & Sons, Inc., 1992

(2) Brumbaugh, Larry J, VSAM - Architecture, Theory, and Applications New York: McGraw-Hill, 1993

(3) Dahl, Dijkstra, and Hoare, Structured Programming, Academic Press, 1972

(4) Gray and Reuter, Transaction Processing: Concepts and Technique, Morgan Kaufmann Publishers, Inc., 1993

(5) LeBert, Joseph J CICS Made Easy New York: McGraw-Hill, 1986

(6) Perry, Timothy T, Melvin R Roy and Richard S Wilkinson Jr. 'A Problem in Online System Development' Interface Winter 1990/91

MEASURING THE COMPLEXITY OF STRUCTURED CODE

R. Wayne Headrick
Associate Professor
Accounting & Business Computer Systems
New Mexico State University
Las Cruces, New Mexico 88003-0001

George C. Fowler
Professor
Department of Business Analysis and Research
College of Business Administration and Graduate College of Business
Texas A & M University
College Station, Texas 77843-4217

Shue-Feng T. Liou
Assistant Professor
National Chenchi University
Taipei, Taiwan, ROC

ABSTRACT

To effectively promote the generation of code that is more easily understood, software complexity metrics must consider both the type of problem being solved and the control structures available for its solution.

INTRODUCTION

"Good" code is generally considered to be code that is written, using structured techniques, in such a way as to make it as easy as possible to debug, test, implement and maintain. The cost of producing such code tends to be reduced because (1) it is written more quickly, (2) it contains fewer defects, and (3) the defects that do exist are more easily identified and corrected. Such code also tends to be relatively more reliable because its understandability facilitates more appropriate, thorough operational testing. Finally, maintenance of such code is also generally less costly because code that is more readily understood is more easily repaired or enhanced.

The question, of course, is not whether "good" code should be written, but rather what constitutes such code. Attempts to address this question have produced a considerable volume of literature related to the topic collectively known as software complexity metrics. Despite the availability of many software complexity metrics, and the obvious potential their application has for aiding in efforts to reduce information system costs throughout the system life cycle, they have gained only minimal acceptance. Kearney et. al. [8] indicate that this lack of acceptance is caused by the absence of purpose associated with their development.

The basic purpose of all software complexity metrics is to provide tools to aid in the quantitative analysis of the complexity of various software solutions to particular problems, with complexity considered to be some measure of the resources expended in the process of interacting with that software in the accomplishment of a given task [2]. In the context of programmer/software interaction, the tasks on which resources are expended include writing, debugging/testing and maintaining code. Thus it is in the identification of the most easily written, implemented and maintained code that software complexity metric development efforts must sustain a major emphasis.

MEASURING COMPLEXITY

Focusing on program size, as indicated by the number of operators and operands in the program, a number of software metrics collectively referred to as software science have been developed to estimate the relative effort required to implement and understand particular programs or code segments [6,7]. As the control structures within a program almost certainly affect the program's complexity, a variety of software complexity metrics have attempted to capture that aspect of the software complexity issue [4,9,11]. Recognizing that the interrelationships among program modules may have an impact on the complexity of the program, other metric development efforts have focused on various facets of program module interaction [10,13,14]. Other studies have suggested the development of a software complexity metric that incorporates software science, control structures and/or module interaction consideration into a single measure [2,15]. While there is substantial evidence that metric value improvements often result from the implementation of some "good" programming practices [1], in nearly all cases, there are also enough examples showing that the metric value can be improved through the use of questionable programming techniques to cause the robustness complexity metrics to be questioned.

Noting that programmer behavior is influenced by the task being undertaken, the ability and experience of the programmer in relation to both the specific task at hand and programming in general, and the environment in which the task is being accomplished, Kearney et. al. [8] assert that the measurement of software complexity must be related to programmer behavior. Viewed in terms of the task and environment that programmers seldom look forward to (i.e., correcting or enhancing another programmer's operational code under a particularly short deadline), "good" code is almost certain to be described as code that is easily read and understood with variable names and logic control structures that are appropriate for the situation. By extension, this same description should also hold when programs are being written and modified.

Soloway and Ehrlich [12] found that experienced programmers develop sets of rules that guide them in their programming efforts, with programs that deviate from those rules tending to cause confusion. Although adherence to a particular set of rules decreases confusion for the original programmer, attempts by another programmer (with his/her own set of rules) to maintain the original programmer's code can cause significant confusion for the second programmer. It is apparent that, when approached from a programmer behavior perspective, what constitutes complex code is relative rather than absolute. As such, measurement of the complexity of a program or code segment must transcend individual programmer behavior and focus on much more basic issues.

Whether measuring program volume, control structures or data/module organization, software complexity metric research efforts are driven by a desire to increase productivity in the development and maintenance of quality information systems. Reducing the cost of information systems development, increasing the reliability of the resultant system, and decreasing the frequency and magnitude of system maintenance activities all ultimately depend on making the system code more easily understood by more programmers.

THE PROBLEM

It has been shown that any logic problem can be solved using various combinations of the basic control structures: sequential, selection and iteration [5]. Sequential control, of course, simply involves passing control from one program statement to the next, and is graphically represented by a directed arc between two nodes representing program statements (see Fig. 1). It should be noted that within a particular module, even statements that invoke externally defined functions and/or procedures are considered to exhibit sequential control. The selection control structure is implemented using the **if**, **if-else** and/or **case** as illustrated in Fig. 2. Finally, the iteration control structure has been implemented in a variety of operational forms (e.g., **for**, **while**, **until**, etc.). Because the differences among the various forms are considered to be minimal, a single representative form, the **while**, has been chosen to represent the iteration control structure in this study (see Fig. 3). It is through the use of these three basic control structures that solutions to even the most complicated of logic problems are built.

----
insert Figures 1, 2 and 3 here

It is obvious that there is often more than one structured solution to a particular logic problem. If so, it is reasonable to expect that one of the possible solutions would be more easily understood (i.e., less complex) than the others. In such a situation, it is also reasonable to expect that the most easily understood solution would be chosen for use. The problem is determining which of the candidate solutions would result in the most easily understood code.

THE EXPERIMENT

The general objective of this study is to explore the relationship between the type of logic problem being solved and the control structures with which is could be solved, and their impact on the understandability of the resultant code. More specifically, it focuses on a particular common logic problem to (1) empirically determine if there are significant differences in complexity among the possible structured solutions to that problem and (2) compare the empirical findings with the results of application of selected software complexity metrics. The particular logic problem chosen to serve as a basis for this investigation, the linear list multi-level control break summarization and reporting problem, has a number of possible solutions that all meet the requirements of structured programming. Even the relatively simple two-level control break problem can be solved through the use of any of the following implementations of structured logic:

(1) linear (un-nested) **if** selection structure,
(2) nested **if-else** selection structure,
(3) nested **while** iteration structure,
(4) **if** selection structure nested within **while** iteration structure, and
(5) **case** selection structure.

To measure the complexity of these structured solutions to the two-level control break problem, the following metrics were used.

(1) McCabe's cyclomatic complexity measure - V(G) [9].
(2) Gordon's clarity estimator - E_c [6].
(3) Nejmeh's acyclic execution path complexity measure - NPATH [11].
(4) Experimental pair-wise comparisons of the various implementations by novice programmers -

Exp(N) (a description of the experiment is provided below).

(5) Experimental pair-wise comparisons of the various implementations by experienced programmers - Exp(E).

The hypothesis underlying the experimental pair-wise comparisons of the solutions was that there would be no significant differences among the complexity levels of the above noted solutions to the two-level control break problem if they are all implemented using good structured code. The experiment was accomplished using two distinct groups of business oriented programmers to evaluate well-written COBOL-85 programs in which identical sets of subordinate procedures (COBOL paragraphs) were driven by main procedures (Figures 4 through 7) containing the different structured solutions to the two-level control break problem. The first group consisted of 84 novice programmers, students in an introductory COBOL programming class who had gained an adequate knowledge of both the selection and iteration control structures, but had not yet been introduced to the concept of either single or multi-level control breaks. The second, experienced, group was made up of 60 professional programmer/analysts with Mobil Oil Corporation's Dallas (TX) Computer Applications and Systems Division, each with a minimum of two years of COBOL programming experience.

Because it has only recently begun to be made available to business programmers (COBOL-85 is the first ANSI standard COBOL to include a statement implementing the **case** selection structure), there is still relatively little knowledge of the case selection structure among business programmers. Because an assumption in the experiment design was that all participants had an adequate knowledge of the control structures being investigated, the implementation of the **case** selection structure shown in Figure 8 was not included in this part of the study.

----

insert Figures 4 through 8 here

Presented with an instrument containing two of the solutions to the two- level control break problem, each participant was asked to use any desired program documentation technique to briefly show the logic of the main body of each of the programs (to ensure an adequate comprehension of both solutions) and then indicate whether one of the solutions was more complex than the other. The resultant pair-wise perceptions of relative complexity were used to compute mean responses for each of the comparisons.

THE RESULTS

The results of the experiment (ref. Table 1) indicate that both the novice and experienced programmer groups identified the linear **if** structure implemented in solution A as being the least complex; and all the participants

----

insert Table 1 here

----

except the experienced programmers who directly compared the nested **while** structure (solution C) to the **while** structure with a nested **if** structure (solution D) identified the nested **while** structure as being the most complex. To view the results in terms of the relative complexity of the individual solutions, the following procedure was used:

(1) Representative mean comparisons between solutions A and B, B and C, and C and D were determined by computing an average comparison along all non- cyclic comparison paths between the solution pairs.

(2) A baseline relative complexity value of 1.00 was assigned to solution A.

(3) Relative complexity values for solutions B, C and D were determined by using the previously computed representative mean comparisons to calculate relative offsets from the baseline value of 1.00 assigned to solution A.

The last two columns of Table 2 detail the empirical relative complexity values Exp(N) and Exp(E) resulting from this procedure. In addition to again identifying the linear **if** structure as the least complex solution to the two- level control break problem, the results also show that both the novice and

----

insert Table 2 here

----

experienced groups ranked the solutions in the same order of relative complexity. Results of the application of the other metrics being considered in this study to the solutions to the two-level control break problem presented in Figures 4 through 8 are also shown in Table 2, and are discussed below.

DISCUSSION

The empirical investigation has shown that both novice and experienced programmers perceived differences in complexity among the structured solutions under consideration. Specifically, both groups found the linear **if** structure to be the least complex, with the nested **if-else**, **if** nested within a **while** and nested **while** following in order of increasing complexity.

Applying McCabe's cyclomatic complexity measure [9] to the various solutions, all of them were found to have a V(G) = 5. Considering that applying each of the solutions to the same data set would result in exactly the same activities in exactly the same order, this result is not particularly surprising. As a measure of possible control flow paths in a program, the McCabe metric is still quite valid. However, it does not have the capability of providing information regarding the comparative ease with which possible solutions to particular logic problems will be understood by programmers.

Gordon's estimator of program clarity E_C was designed specifically to quantify the mental effort required to understand a program [6]. As such, there was some expectation that a direct relationship might exist between E_C and the measures of programmer perceived relative complexity Exp(N) and Exp(E). Table 2 shows that E_C correctly identified the solutions perceived to be the most and least complex, A and C respectively. However, it reversed the rankings of the two other solutions as compared with Exp(N) and Exp(E).

Further analysis of the formulation of E_C indicated that it is particularly sensitive to changes in the number of unique operators and, to a lesser extent, the total number of operands. As the various solutions were prepared under the most general of multi-key sequenced linear list data set assumptions (i.e., a minor key change does not have to accompany every major key change), it was noted that making a simplifying assumption requiring such a change would allow the removal of the sixth line of code in solution C. A relatively minor change that does not alter the solution's control structure, it would not be expected to have a significant impact on the solution's perceived relative complexity. Because such a change would allow the removal of one unique operator, two total operators and two total operands, it would result in a reduction in E_C(solution C) from 2221.7 to 1687.4, thereby identifying it as being significantly less complex than any of the other solutions. Although it conforms best with the empirical Zxp(N) and Zxp(E),
lexical token based metrics such as E_C are not able to consider control stuctures in their computations and, as such, are severely limited in their ability to measure programmer perceptions of code complexity.

Designed to measure a program's execution path complexity, the NPATH metric [11] is a function of both a program's acyclic execution paths and the logical statements that direct the execution of those paths. Related directly to program control structures, NPATH could be expected to rank solutions A through D in the same order as Exp(N) and Exp(E). Instead, it ranked them in exactly the reverse order, with the linear **if** structure being rated the most complex.

An analysis of the NPATH calculations indicates that they are slightly biased in favor of nested control structures, while other studies have found that the use of nested control structures can increase the perceived complexity of a program. Although modification of NPATH calculations can readily change its bias as necessary, it does not consider the suitability of particular control structures for use in solving specific logic problems. As such, it does not appear to be appropriate for use in measuring programmer perceptions of code complexity.

CONCLUSIONS

It has been recognized that as every programmer gains experience, he/she develops a repertoire of programming rules to guide the writing of code that will subsequently be more easily understood by that programmer. The challenge is to identify those programming rules that tend to produce code that is easily understood by most programmers. Software complexity metrics are developed to provide quantitative measures of the programmer resources expended on writing, debugging/testing and maintaining code. As such, they should be able to aid in the identification of programming rules that promote the generation of code that tends to decrease those expenditures - code that is more easily understood by all programmers.

Excluding some individual differences of programmers, program control structure and problem context seem to have the most impact on how well the program can be understood by most programmers. It is apparent that software complexity metrics that do not include consideration of the control structures used in a program are not able to accurately predict the relative ease with which the program can be understood. In addition, even those metrics that consider program control structures, but do not consider them in the context of the logic problem being solved, are unable to accurately reflect programmer perceptions of relative complexity.

The pair-wise comparison of alternative solutions to the two-level control break programming logic problem accomplished in this study identified the linear **if** control structure as the least complex of the structures considered, with the nested **if-else** control structure ranked second. As such, the use of more "elegant" control structures appear to accomplish nothing other than adding unnecessary complexity to this problem's solution. Exit interviews with the experienced programmers served to reinforce the need to consider the type of problem being solved when attempting to measure the complexity of proposed solution. Their comments indicated that, even though they generally approved of using the iteration **while** control structure to ensure processing of all the records in a data set, they considered the actual control break processing decisions to be linear record-by-record decisions best described by simple **if** control structures.

It is important, therefore, to recognize that the results of this study do not imply that a generalization such as "use only the linear **if** control structure" should be a programming rule. What they do indicate is that to accurately assess the relative ease with which a solution to a particular programming logic problem is understood, a software complexity metric must consider both the problem being solved and the control structures with which is could be solved.

NOTE: Tables and Figures will be provided at ISECON '94 Session.

REFERENCES

1. Baker, A.L. and Zweben, S.H. The use of software science in evaluating modularity concepts. IEEE Trans. Softw. Eng. SE-5, 2 (Mar. 1979), 110-120.

2. Basili, V.R. and Hutchen, D.H. An empirical study of a syntactic complexity family. IEEE Trans. Softw. Eng. SE-9, 6 (Nov. 1983), 664-672.

3. Bohm, C. and Jacopini, G. Flow diagrams, turing machines and languages with only two formation rules. presented at the 1964 International Colloquium on Algebraic Linguistics and Automata Theory in Jerusalem, Israel, and printed in Commun. ACM 9, 5 (May 1966), 366-371.

4. Chen, E.T. Program complexity and programmer productivity. IEEE Trans. Softw. Eng. SE-4, 3 (May 1978), 187-194.

5. Evangelist, W.M. Software complexity metric sensitivity to program structuring rules. J. Sys. Softw. 3, (1983), 231-243.

6. Gordon, R.D. Measuring improvements in program clarity. IEEE Trans. Softw. Eng. SE-5, 2 (Mar. 1979), 79-90.

7. Halstead, M.H. Elements of Software Science. Elsevier North-Holland, New York, 1977.

8. Kearney, J.K., Sedlmeyer, R.L., Thompson, W.B., Gray, M.A. and Adler, M.A. Software complexity measurement. Commun. ACM 29, 11 (Nov. 1986), 1044- 1050.

9. McCabe, T.H. A compexity measure. IEEE Trans. Softw. Eng. SE-2, 6 (Dec. 1976), 308-320.

10. McClure, C.T. A model for program complexity analysis. Proc. 3rd Int. Conf. Software Eng., Atlanta, GA, May 1978, 149-157.

11. Nejmeh, B.A. NPATH: A measure of execution path complexity and its applications. Commun. ACM 31, 2 (Feb. 1988), 188-200.

12. Soloway, E. and Ehrlich, K. Empirical studies of programming knowledge. IEEE Trans. Softw. Eng. SE-10, 5 (Sep. 1984), 595-608.

13. Woodfield, S. Enhanced effort estimation by extending basic programming models to include modularity factors. Ph.D. dissertation, Purdue Univ., 1980.

14. Yau, S.S. and Collofello, J.S. Some stability measures for software maintenance. IEEE Trans. Softw. Eng. SE-6, 6 (Nov. 1980), 545-552.

15. Zolnowski, J.C. and Simmons, D.B. Measuring program complexity in a COBOL environment. Proc. 1980 Nat. Comp. Conf. 49, 757-766.

Education at the University of Veszprém, Hungary, with an Emphasis on IS Education

Dr. Zoltán Gaál
Dr. János T. Füstös
Department of Engineering Management
University of Veszprém
Veszprém, Hungary 8200
36-88-422022 X334 fustos@miat0.vein.hu

Dr. Laszlo A. Pook
Computer Informations Systems and Management Science
Metropolitan State College of Denver
Denver, CO 80217-3362
(303) 449-8604 pookl@mscd.edu

ABSTRACT

University education in Hungary has some centuries old traditions. An examination of the university education system in Hungary, specifically at the University of Veszprém, leads to observable strengths and weakness. Building on the available strengths of a high quality student body with strong technical background, an emphasis on theory, and a fine reputation can be used to overcome many of the weaknesses caused by lack of funding, experience, and years of government interference. The IS curriculum in particular is found to contain interesting cultural features.

History of the University of Veszprém

Church records dating to the 13th century show that the first university North and East of the european Alps was founded in Veszprém, Hungary, around 1276. As with most universities it did not enjoy continuous existence. Not until the founding of some of the major universities during the 18th century did they survive to this day.

University governance was at first in the hands of the churches that founded the institutions. By the late 1800's, when many universities were taken over by the central government, governance was turned over to university administrations. With liberalization, the university system flourished and produced some the nation's many Nobel laureates. Research and education were joined to complement each other.

After the communist takeover in 1948, the government centralized university education on the Soviet model. PhD granting rights of all universities were eliminated and the function was turned over to the Hungarian Academy of Sciences. This remained the status quo until 1993. The curriculum was re-designed and placed under the Academy's control. All faculties were purged of liberal thinkers and, department leadership was placed in the hands of Soviet trained academicians and educators. Furthermore, education and research were separated according to the Soviet model. Research became the domain of the Academy of Sciences. Though research activities continued at universities, the work was performed by non-teaching academics under Academy supervision.

The beginnings of the 1956 revolution against communist oppression can be traced to the universities. When the Soviet troops put down the revolt, 334 faculty members and 2879 students were forced to flee. Approximately another 100 faculty members were purged, some even executed (Szogi, 1994). Centralization was complete.

Hungary's experience with centralization in most aspects of the economy was dismal at best. The process was partly reversed in the early 1980's and excelerated after the 1989 fall of the communist government. University autonomy returned, and by 1993 deans and university presidents regained control of their institutions. Budgetary support, however, still depends on government largess.

The university at Veszprém dates from 1948 when it was founded as a branch of the Technical University of Budapest. The University of Veszprém gained university status in 1951 specializing in chemical engineering. In 1991 teacher education was also added. The university's chemical engineers, due to their rigorous education and analytical abilities, were much sought after by industry. An industrial engineering program was instituted within chemical engineering in 1973. An essential component of the program are the areas of quality management and maintenance.

History of the Management Program

Before 1989 when the communist leadership gave up governance of the country, ideas of planning, decision making, information systems, markets, and competition were foreign to most Hungarian universities. This was primarily due to a lack of any need for such capitalistic ideas in the centralized economy. Since then, however, as the nation moved onto a market based economy need arose to train future leaders for industry with managerial, albeit business background. The university's first few moves into business education were tentative. During 1991 and '92 they offered courses in quality control with a strong business bias.

As demand for the business aspect of the engineering program became apparent, the university instituted an engineering management program in 1992. The program is aimed at engineers with aspirations toward business leadership. The curriculum was designed to augment the technical and analytic engineering curriculum with courses in economics, accounting, finance, management, strategic management, marketing, and information systems.

Language training has always been an integral component of Hungarian universities. Graduates maintain high language competence in at least one language, usually in English or German.

The Present

Hungarian universities are organized around areas of specialization rather than around degree granting disciplines. There are sixteen specialty areas within the School of Engineering at Veszprém. These specialty areas produce six diplomas in Chemistry, Materials Engineering, Information Technology, Chemical Engineering, Environmental Engineering, and Engineering Management. Engineering Management is the topical concentration of the Department of Business Management and Economy. There are two emphasis areas within Engineering Management, these are Production Management and Entrepreneurship. The School of Engineering enrolls 1,600 students at present. Of these, the incoming first year class of Engineering Management for 1994 is already 152 students strong. This is an increase from 75 in 1993.

The apparent dynamic growth parallels the country's move toward a market economy and an increasing interest in management principles. It is also the only engineering management program at the university level in the country. According to a recent survey some 80% of graduates from the engineering program at Veszprém are now working in management positions. Furthermore, the emphasis on the combination of a management program with engineering is rooted in the local experience that the outputs produced by most business firms are more likely goods than services. Until very recently this, was the case. While this view is likely to change in the future, the present position is not uniquely Hungarian. Other counties in the region hold similar positions. An overview of the curriculum demonstrates how this duality is accomplished at Veszprém University.

The Curriculum

The degree program is of five years, ten semesters, in duration. It is a tightly designed, full time program, where students essentially study the same courses, excepting electives and emphasis areas. The program is capped by a researched, written thesis. At the end of the ten semesters students receive a diploma equivalent to a masters degree in US terminology.

All together students complete what is roughly equivalent to 180 credit hours of work. Of the 180 credit hours 43 hours are electives in engineering subjects, economics, and management. Beyond these 180 credit hours students at Veszprem take 36 hours of foreign language and 18 hours of physical education. The courses according to the Hungarian system are classified into fundamental, commercial, management, and technical subjects. A loose correspondence may be established with a US system in terms of general studies, general business core, information systems, engineering core, and major area. Table 1 represents an attempt at translating between the US and Hungarian systems.

The Information Systems Curriculum

The information system (IS) curriculum at Veszprém U. consists of three required courses. Two courses, taught as part of fundamentals, which students take during their first year at the university, and the third one, a two-credit hour course titled Management Information Systems (MIS) they take during their fourth year for a total of 8 credit hours of IS and MIS courses. This sequence is similar to many US institutions in that beyond an introductory course students in a school of business take a management information systems course.

Course descriptions, translated from the Hungarian, are as follows (the two fundamentals are listed first):

A4. COMPUTING TECHNIQUES I. (2 credit hours)

Hardware fundamentals: theory of digital operation, components of personal computers. Concepts and application of micro, mini, and mainframe systems. Data fundamentals. Instruction and computation concepts. Software fundamentals: operating systems. Principles of disk operating systems (DOS), system commands, files, instruction sets.

A10. COMPUTING TECHNIQUES II. (4 credit hours)

Software selection guide lines, factors to be considered. Utility-k (PCT, Norton). Text editors (Norton, Ekszer, Word). High level programming languages; language selection, syntax, semantics, translations, word processing, running, error handling, algorithms, flow charts, pseudo code. Language elements, commands, command structures, data structures, data handling.
Disk files: file types (text and record types). Use of monitors and printers. Program and data segmentation. Unit structures and their use. Graphics. Programming in Pascal.

TABLE 1

Credit Hours in the Engineering Management Program

	Hungarian Classification					
US System	Fundamentals	Commercial	Management	Technical	Electives	Row Totals
Gen. Stud.	29		2	2		33
Bus. Core	8	20	22			50
Inf. Systems			2			2
Technical	4		4	27		35
Emphasis		2	7(1) 12(2)	5(1)		14
Electives					43	
Col. Totals	41	22	37(1) 42(2)	34(1) 29(2)	43	177

(1) production management emphasis; (2) entrepreneurship emphasis

V8. MANAGEMENT INFORMATION SYSTEMS (2 credit hours)

Definitions of information, components of information systems, modeling of information systems. Data, data models, data bases. Manual and automated information systems. System development techniques, project management. Organization and management of computerized information system projects. Case studies and assignments. Database organizational concepts.

Differences can be observed between US and Hungarian institutions in two areas here. One is content and the other is hands-on experience. While the introductory courses in Hungary are technical, and in many ways theoretical, students get only a limited view of business applications and the manner in which the business world uses computers for information processing. This view of "theory first" prevails in much of higher education here. Hands-on computer use, and programming are used to explain the operation of the computer rather than its use in business problem solving. Students in the Management Information Systems course do not use computers, rather they apply their learned knowledge in solving conceptual problems.

Referring to the MIS course content above, it may be observed that much of the subject area is what is popularly referred to in the US as transaction processing systems. The theory based presentation does not allow for a behaviorist approach to MIS and so a need based or organization behavior oriented approach to MIS development is not evident.

The following is the formal course content for the MIS course:

1. Alternative interpretations of information. 2. Concepts and elements of information systems. 3. Information system models. Data modeling basics. 4. Data and data sets. 5. Organization functions and computer aided information systems. 6. Traditional and computer aided information systems. 7. Computerized information system development in the company. 8. System development and organization development. 9. Methodology of computerized information system development. 10. The role of user, programmer, systems analyst in the design and development process. 11. Computer related jobs. 12. The place of systems analyst in the computer center. 13. Case studies. 14. Management of information systems projects. 15. Basic data-base concepts.

Observation of the present course suggest that fundamental technical concepts are stressed and emphasis is primarily on operating information system characteristics and design. Decision makers are viewed as needing detailed and precise internal data for decision making, and those designing systems for higher levels in the organization are expected to know system details rather than match system functions to managerial styles. With a few exceptions, such as a course in business simulation, computers are only occasionally used in the class room. This of course is partly due to the few computers available, and partly the result of the belief that computers are tools to be mastered on site as needed. The "theory first" approach to teaching results in some interesting sequencing of courses. Case in point is a course in strategic management containing among other things managerial working and decision making styles (Mintzberg, Keen and Morton models, etc.) is offered subsequent to the MIS course.

Hardware Used in IS Education

In the two computing fundamentals courses (Computing Techniques I and II), students work with PC operating systems and learn to program in Pascal. Their experience with mini or mainframe computers is non-existent to minimal. Details of hardware design, structure, operating conditions, and hardware technology are examined with an emphasis on personal computers.

Network use is similar in that only those students who involve themselves in courses using networks receive network experience. One such course is simulation in management. The course is conducted over Internet in cooperation with the University of Wisconsin at Madison. Network experience consists of trading strategies over the Internet network. Telecommunication technology is taught from a technical standpoint by the Informatics (computer science) branch of the engineering school. Students majoring in engineering management are encouraged to take other elective courses in engineering (see Table 1). Some will take a course in telecommunication technology.

Software Used in IS Education

Software used is primarily PC based software, DOS, Pascal, dBase3, and spreadsheets. At present, little to no emphasis is placed on productivity software use in academic courses. DOS operating systems and programming concepts are studied in detail as the course description above illustrates.

On occasion, software is used during class periods to demonstrate concepts presented in class. This is primarily because of the theory first principle applied in education, and only secondarily due to lack of equipment. Some demonstration equipment is available. However, there is certainly a lack of freely available equipment throughout campus to carry enough inertia to motivate further familiarity with computers as tools.

Problems Encountered

Information system use as a tool in business decision making needs to be integrated into the curriculum. Some of the components are already there. For example a series of courses are offered as elective subjects which, if required early in the academic program, may become eventual motivators for effective computer use later on in upper division subjects.

These courses are mostly productivity software based such as VV9. Databases with SQL (2 credits), and VV10. Spreadsheets, (1 credit). Proper mix of theory, practice, and applications to real life problems could assist at acceptance of the technology by developing skills early, overcoming anxieties, and demonstrating the usefulness of the technology. A course, titled VV 8. Networks (2 credits), which is also an elective course at present, could be used both as a hardware and software course to develop LAN and WAN concepts from the user standpoint. The department can rely on the already existing network courses offered by the Informatics branch for technical training.

The remainder of the management curriculum needs an infusion of computer technology as well. Students with increasing computer skills coming up the years will no doubt demand further computer use in the class room. This is paramount since the economy itself is rapidly computerizing as evidenced by both government and private enterprise turning toward information technologies.

Of course technology does not come cheep anywhere and Hungary is not a wealthy nation. The university budget is 100% government funded. Allocation is based on student enrollment, sitting faculty numbers, and fundamental operating needs. Students receive a stipend from the government based on their entrance exam and/ or high school performance. The stipend covers their education as well as part of their living expenses. For the 1994 budget year average stipend per student is budgeted to be about $650 pert year with an additional $100 per year after September 1994 (Polonyi, 1994). With an average personal income of $3,264 (1 USD=100 HFT, 3/13/1994) much government assistance is necessary. There is, in theory, private tuition, but nobody can afford it in Hungary, and few foreigners can attend Hungarian universities because of the language barrier. The university has some external funding sources but these are minimal, and just enough to cover outstanding operating expenses not in the annual budget.

If universities manage to move at least part of the instruction onto a common foreign language, such as English, and attract wealthy foreign students not only would the institutions gain financially but the foreign infusion would be an enriching experience for all parties concerned.

Traditionally, students entered the university system out of high school and continued until they received their diplomas and a job, or continued toward a PhD. Once in the labor force, rarely if ever would students return for advanced degrees. With recent liberalization universities have been experimenting with advanced, post graduate, training programs followed by certification. Whether these certificates will be accepted as advanced degrees is yet to be seen. The advanced training programs will hopefully serve as the forerunners of adult education and advanced degree programs. They certainly offer additional revenue and could become an important source of revenue if their worth is proven and if businesses continue to shoulder the expense instead of the government.

Future Plans

With the available faculty, apparent student interest, and expanding national market economy it would be natural for the University of Veszprem to expand engineering management education into a full business program with strong ties to technical albeit engineering training.

Further plans are to evolve from a department of management an eventual school of business at Veszprém with Hungarian as well as overseas accreditation. Hungarian accreditation is necessary in order to acquire PhD granting authority in business. Overseas accredition is needed for successfull student exchanges as well as for international recognition and eventual private funding. There are strong european traditions in Hungary which dictate a scientific basis to business and management. With the equally traditional 5 year program in place Hungarian universities can continue producing outstanding college graduates with both business and engineering backgrounds. It is likely that if funding becomes available, the University of Veszprém's engineering management program will expand into a business program with strong ties to technical disciplines. With autonomy regained and the possibility of students and private businesses shouldering more of the tuition burden there is even talk of a graduate program in business.

A home grown masters degree program in business, whether MBA or MS program, would have advantages over the present hodge-podge of Western institutions offering everything from first class MBA education costing thousands of dollars to third rate diploma mills. A graduate degree program developed locally and staffed mostly from Hungarian institutions would have the trust of local businesses, would be affordable, and best of all would convey the Hungarian experience rather than that of another national and business culture. It is one thing to learn from the experience of another but it is quite different to also mimic it.

References

Polonyi, Istvan: "A Felsooktatas Lehetosegei az 1994. Evi Koltsegvetesi Torveny Tervezeteben," Magyar Felsooktatas, Ministry of Art and General Education, July, 1993. ("Alternatives for Higher Edcation under the 1994 Budget Plan," Hungarian Higher Education).

Soos, Lorinc (Ed.): Monthly Bulletin of Statistics, Kozponti Statisztikai Hivatal, Budapest, Jan. 1994.

Szogi, Laszlo: "A Magyar Felsooktatas a II. Vilaghaboru Befejezesetol a Napjainkig, I-III," Magyar Felsooktatas, Ministry of Art and General Education, 7-10, 1993. ("Hungarian Higher Education Between WW2 and Today," Hungarian Higher Education).

Information Systems Study in Hong Kong: An American Ex-patriates View

John F. Schleich, PhD
Department of Information Systems
City Polytechnic of Hong Kong
E-mail ISJACK@CPHKVX.CPHK.HK
Phone: (852) 788-8493 FAX: (852) 788-8694

Abstract

Information Technology (IT) programs in Hong Kong are designed to meet the needs of Hong Kong. The IS courses in Hong Kong are very popular and competition for placement is very keen. Hong Kong has been administered by British expatriates for over 150 years and has taken on a United Kingdom look in its educational system. This paper focuses on the Information Systems Courses at The City Polytechnic of Hong Kong.

Introduction

On June 30th, 1997 at midnight, the Union Jack, flying for more than 150 years over the Hong Kong Governor's mansion, will be replaced by the flag of China. One scenario for the aftermath is that all has been negotiated in the "Basic Law" and Hong Kong will remain uniquely Hong Kong. At the other end of the spectrum is the scenario where the new government replaces all the old institutions with ones possessing total revolutionary zeal. The reality may well lie somewhere between the two scenarios. The oft heard saying that "China needs Hong Kong as Hong Kong" is accurately reflects the situation. This paper discusses one part of current reality - the tertiary institutions' Information Systems programs - that may well escape unscathed in the 1997 changeover.

The closest relative to the Colony of Hong Kong is perhaps Singapore. While Singapore is a city-state responsible to no other authority, Hong Kong is a city-state currently under British Authority, but soon to become an integral part of mainland China. While Singapore often boasts of its access to the China market because of its mainly Chinese population the country is still a sovereign nation, Hong Kong is different, it is not a sovereign nation and the 96% Chinese majority, consider themselves Chinese -- that is mainland Chinese.

The Educational Structure of Hong Kong

The educational systems in Hong Kong must be looked at in light of the cultural aspects of Hong Kong. The traditional reverence for education is entwined with the British influence to produce a unique educational system. There are two readily apparent factors that have a profound impact on Hong Kong's Educational systems. The first is the intense desire to get as much education as possible that is instilled in the population and the second is the willingness of Hong Kong to spend money on education. No other program has a larger share of the Hong Kong budget. Note: in 1993 Hong Kong had a $2.4 billion dollar revenue surplus, some of which is being ear marked for education. This surplus reflects that Hong Kong does have large revenue to support the educational system.

A look at the components that make up an Institutions of Higher learning - Students, Faculty, Facilities, Curriculum, and Administration shows the intertwining of the cultural aspects and money.

Students: The competition for placement in degree programs (and in diploma) programs is very keen. Students finishing their secondary schooling have first crack at placement in advanced programs. This placement is supervised by the University and Polytechnic Grants Committee (UPGC) which is shown as a major educational subdivision under the Governor in Figure 1. Students are asked to state their ranked choices for twenty programs offered by the UPGC institutions. The program coordinators are then given a list of students who have indicated that program as one of their choices. The "program" then ranks the students. UPGC, using student rank and "program" rank, selects the students to be matriculated and makes each qualified student one offer. Note: the offer will be for the student's highest choice for which the student was qualified and the program had an opening.

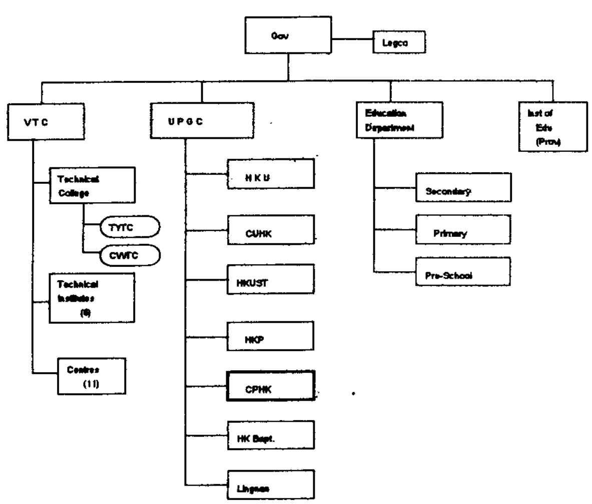


Figure 1. Hong Kong Education Structure[1]

Faculty: The faculty is composed of a combination of locally qualified people and ex-patriots, recruited for posts when no local qualified individuals are available. Ex-pats make up about 30% of the current teaching staff with percentages varying from discipline to discipline. Salaries are relatively high with Australians and British ex-pats doubling and sometimes tripling what they could earn in their home country. Salaries are not guided by market forces as they are in the US and Canada, but all disciplines have equal pay. Although, in some fields, faculty may be bought in higher up the scale. Current practice at most institutions is to require a PhD for appointment as faculty.

Facilities: As can be noted in Figure 1, there are seven tertiary institutions under the UPGC. The oldest of which are Hong Kong University (HKU) and Chinese University (CU), the newest is the Hong Kong University of Science and Technology (HKUST), whose name invokes the perceived Hong Kong educational need. Both The Polytechnics are fairly recent (70s &80s) additions to the tertiary scene. Hong Kong Baptist (HKBC) and Lingnan Colleges are also state schools that were, at one time, private. The newer schools, and the colleges are building new campuses that have state of the art teaching classrooms, laboratories, and computer facilities.

Administration: The ratio of administration to faculty (or students) is heavier that in the US and this may be due to the former abundance of qualified labour. All administration activities seem to be covered by written policies which appear not to change but rather are used as " .. we can't do that since we've never done it before" as well as "... that's the policy". More and more decision-making administrators are local rather than ex-pat.

Curriculum: The variety of curricula offered at the seven UPGC institutions is controlled by the UPGC to reflect the future needs of the Hong Kong community. New courses are added and old courses are revalidated (or replaced) only after determining the need and structuring the course to meet this need. The validation process is quite comprehensive and after being done at the departmental and faculty level is sent to the University and the UPGC where outside experts

[1] Contributed by Dr. Ken Yiu, Accountancy Department, Hong Kong Polytechnic

are given the opportunity to comment on the proposals. The single biggest hurdle is the "needs of Hong Kong". While in the US college graduates often discuss their "degrees", in Hong Kong the graduate refers to his/her "qualification". This points to a cultural difference which impacts overall tertiary education.

Students are enrolled in a specific course and his/her schedule for the course duration is set at enrolment. Students rarely are allowed to drop back a year or even more rarely to change their field of study. Students will have the same classmates in every module of instruction for the duration of the program.

Figure 1 shows a simplified block diagram of the governmental structure of the education system in Hong Kong. The main task of the figure is to highlight the Tertiary component of Hong Kong's educational system. The three main components of the Hong Kong educational system are the Vocational Training Centre (VTC), the University and Polytechnics Grants Committee (UPGC) and the Education department. A more recent addition (not shown) is the Open Learning Institute which specialises in distance learning.

In figure one the educational components are shown directly reporting to the governor. A better view might well be to place them under the Legislative Council (Legco), an elected body similar to the US House of Representatives but with much less power or authority.

The VTC has two Technical Colleges and six Technical Institutes. The technical colleges are roughly equivalent to Community colleges in the United States and conduct courses leading to a "Diploma" and "Higher Diploma". These are both non-degree programs designed for students who have completed Form 5 (approximately US high school). The UPGC covers the seven state degree granting institutions as shown in Figure 1. These institutions have more that 66,000 students enrolled (Roberts, 1992). Two of the institutions have separate departments of Information Systems, CPHK (see figure 2) and HKUST. The Hong Kong Polytechnic (HKP)

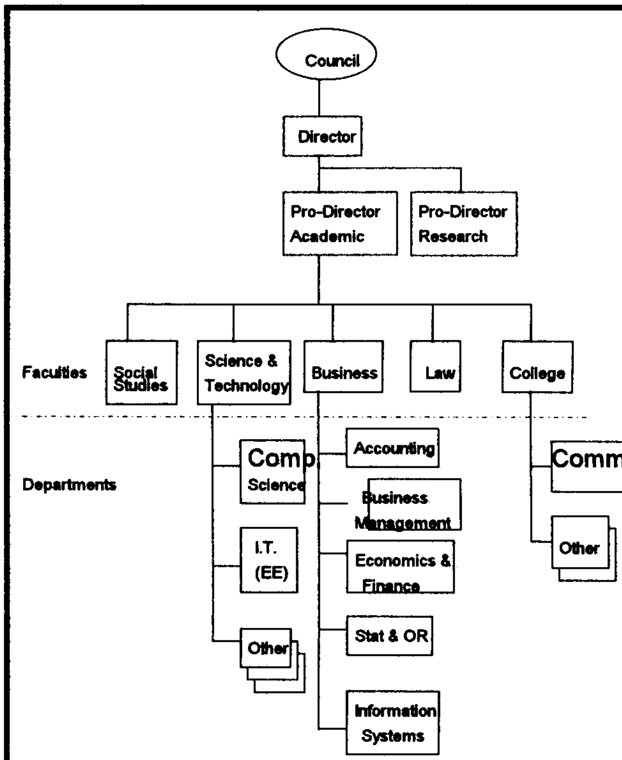


Figure 2 City Polytechnic Structure Overview

has a Department of Computer Studies located in the Faculty of Business with Information Systems and Computer Science streams. HKU and CU tuck Information Systems into another department and have only a small faculty. HKBC's Computer Science Department houses Business Computing and resides in the Faculty of Science and Technology. Lingnan college has a separate area in Information Systems in the Business Division.

CPHK

CPHK has a total area of over 160,000 Square metres to house the academic, administrative and support functions. The total student population is projected to be over 15,000 (FTE) for 1994 (Postgraduate, 1994). The Polytechnic is divided into 5 faculties each headed by a Dean or Principal (see figure 2).

Also shown in figure 2, the Information Systems department (IS) at CPHK is a part of the Faculty of Business. IS separated from the computer studies department in 1990 and the parent department is now the Computer Science Department in the Faculty of Science and Technology. There are two other computer Faculty related degree programs at CPHK: the Information Technology Bachelors of Science in the Electrical Engineering Department and the Bachelor's of Science in Computer Mathematics given by the Mathematics Department. Both of these programs are in the Faculty of Science and Technology. These are shown in figure 3.

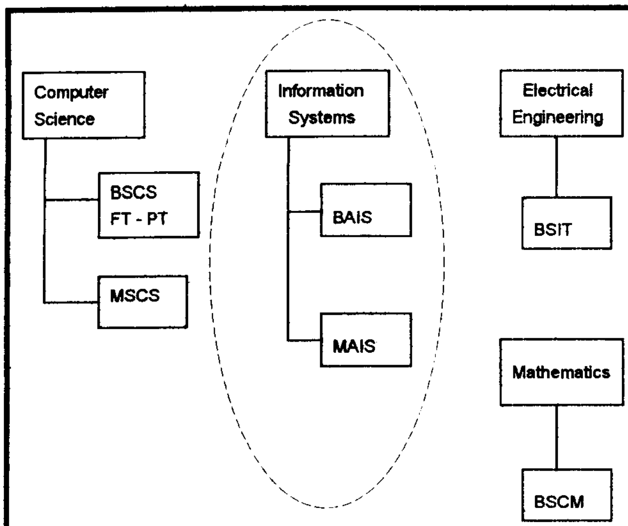


Figure 3 CPHK Computer Based programs

Degree programs at CPHK are generally three year programs. Entry into these programs normally goes to applicants who have finished two years beyond high school, called Advanced or "A-levels". The A-level examination results are critical to the admission process to degree programs. The A-levels may also provide a "liberal arts" element to overall tertiary education which appears to be missing in most Hong Kong curricula.

Degree program are either full time, part time or "sandwich" (which requires the student to work full time for a period during the degree program, similar to US co-op program.) The sandwich course will be stretched out accordingly.

Part time programs are exclusively for more mature students and an entry requirement for many programs is that the student be employed full time in the area of the "qualification". Students in the part time mode are expected to attend classes three nights a weeks from 6:30PM to 9:30PM. In some instances the student may need to attend four nights or even five. Most part time students are in their late 20s or early 30s, although there are some older students in the part time programs. Generally the employer does not pay or reimburse for part time studies.

Part time programs at the bachelor's level generally last for six years and the requirements are identical to those of the full time program. The part time graduate programs last for three years

Information Systems Department

The Department of Computer Studies was broken into two departments in 1990. These were the Department of Computer Science, which was housed in the Faculty of Science and Technology and the Department of Information Systems which became one of five departments in the Faculty of Business. The charter faculty came from the computer studies department and the IS department was initially a service department for both Degree and Higher Diploma courses spread throughout all the CPHK faculties. This year also saw the Information Systems department begin it's own degree program the Master of Arts in Information Systems (MAIS).

In 1991, a new "faculty", *the College of Higher Vocational Studies,* was created and took over diploma and higher diploma courses, leaving all degree teaching to the other four faculties. This was followed by the second program, the Bachelor of Arts in Information Systems (BAIS). These two courses epitomise the teaching of data processing and information systems in Hong Kong.

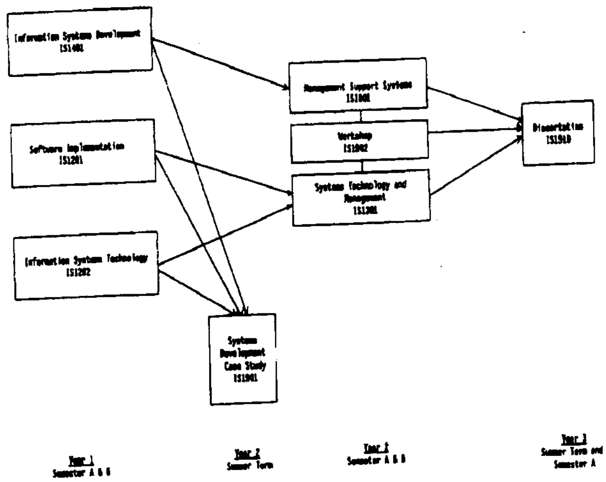


Figure 4 MAIS Program (from MAIS brochure, undated)

The successful implementation of both programs was due in a large part to the computer laboratories in IS. There are currently four laboratories in the department manned by a support staff of 7 hardware or software technicians. Three of the labs are teaching labs, while the other is a project lab for BAIS students in their final year. The two IS degree programs have first priority for use of the IS labs.

The MAIS Program

The MAIS is an evening program with a normal intake of 40 students per year, although a one time intake of 60 students entered the program last year. The degree is called a conversion degree in that students come with Bachelor's

Module	Year	Subject	Sem	Lecture	Tutorials	Labs
IS1401	1	Information systems development	A B	30 15	15 15	15 15
IS1201	1	Software Implementation	A B	15 15	15 15	15 30
IS1202	1	Information Systems Technology	A B	15 15	(15) (15)	(15)* (15)
IS1901	Sum	Systems development Case Study		NONE	NONE	NONE
IS1801	2	Management Support Systems	A B	30 30	15 15	15 15
IS1301	2	Systems Technology and Manage ment	A B	30 30	15 15	15 15
IS1902	2	Workshop**	A B	NONE NONE	NONE NONE	15 15
IS1910	3	Dissertation	A B	NONE NONE	NONE NONE	NONE NONE

* A total of 15 hours Tutorial or Lab
** Used for coursework assessment for both IS1301 and 1801

Table 1. MAIS Program Coursework

degrees from other than computer areas. The program is a three year program which culminates in the third year in a Master's dissertation. The block diagram of the program is shown in figure four.

The stated aim of the course is "...to produce graduates who are able to develop end-user application systems, to work closely with computer professionals in the development of corpo

rate information systems, to contribute effectively to the development of strategic systems plans and to evaluate information systems technology for strategic advantages."

As a part time program the duration is three years, as is illustrated in figure 4. There will normally be 120 MAIS at CPHK at any given time, forty in each year. As can be noted in the block diagrams, the final year is devoted to an MAIS dissertation. Students who opt not to do the dissertation are awarded a Post-Graduate Diploma in Information Systems. This only requires that the student successfully complete the "taught modules". A listing of the courses with a brief description can be found in Table 1.

BAIS (Honours) Program

The BAIS program is a full time program with a normal intake of 90 students per year. This means that the normal complement of BAIS students in the department is 270. The actual intake each year will vary depending on CPHK needs. The block diagram of the program is shown in Figure 5. Note that no modules for full time courses are taught in the summer. The introduction to the course states ...the context for the BAIS course "*is the efficient development of information systems for organizational effectiveness. The information systems professional produced by this course has a sound understanding of organisations, their functioning and behavior as complex socio-technical systems, and is therefore able to apply information technologies*

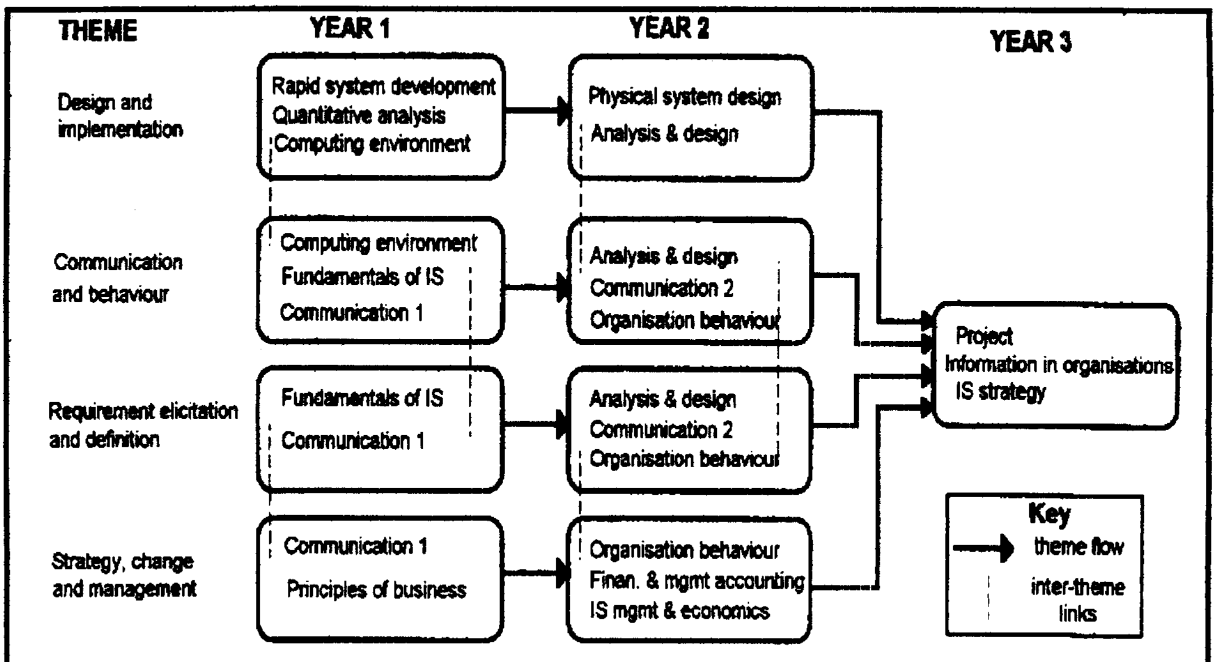


Figure 3 BAIS Program (From Jordan , 1994)

	Year	Subject	Semester	HOURS Lecture	HOURS Tutorials	HOURS Labs
IS3105	1	Fundamentals of IS	A	30		15
			B	30	15	
IS3205	1	Rapid Systems Development	A	30	15	15
			B	30	15	15
CS0113	1	Computing Environment	A	30	15	
			B	30	15	
AR0731	1	Quantitative Analysis	A	30	15	
			B	30	15	
BM2402	1	Principles of Business	A	30	15	
			B			
EN1245	1	Communications for Computing I	A			
			B		45	
IS3301	2	Physical Systems Design	A	30	15	
			B	30	15	
IS3401	2	Systems Analysis and Design	A	30	15	
			B	30	15	
IS3501	2	IS Management and Economics	A			
			B	30	30	
AC4110	2	Accounting for Financial Decisions	A	30	15	
			B			
AC4250	2	Managerial Accounting	A			
			B	30	15	
SS1201	2	Work Behavior in Organisations and Society	A	30	15	
			B	30	15	
EN1246	2	Communications for Computing II	A		45	
			B			
IS3111	3	Information in Organisations	A	30	15	
			B	30	15	
IS3601	3	IS Strategy	A	15	30	
			B	15	30	
IS3901	3	Project	A	NONE	NONE	NONE
			B	NONE	NONE	NONE

Table 2. BAIS Program Coursework

and methods to enterprise problems and to perceive opportunities to use information systems for competitive advantage. It is further stated *...The course has been designed to provide adequate supporting cover of the underlying disciplines and full treatment of information systems methods, while still permitting the student to follow particular interests through elective topics relevant to the main thrust of the course.* The course was designed to meet the requirements of the British Computer Society (BCS). This Society is more Computer Science and technically oriented. The CPHK BAIS course failed to gain exemption from the BCS from examinations that are generally required for membership in the organization. The main reservations of the BCS was their doubt of the engineering underpinning of the BAIS program. *Note: Both the CPHK BScCS and the MAIS were granted some exemptions.*

The students take a total of 99 semester hours in the course or which 54 can be classed as IS modules. As with the MAIS the modules are a mixture of lecture, tutorials, and laboratories. This is shown in Table 2.

Electives	Subject Area
IS3001	Topics in Information systems
IS3521	Information Systems Auditing
IS3701	Knowledge Acquisition for Expert Systems
IS3711	AI and the Design of Expert Systems
IS3801	Decision Support Systems
IS3811	Manufacturing Information Systems
Serviced Electives	**Number of Courses**
Accounting	4
Chinese	3
AR	1
Management	10
Comp Sci	1
Finance	1
English	1
Social Admin	3
Social Studies	1

Table 3 BAIS Electives

Additionally, unlike the MAIS, the students are given a choice of electives. The electives are offered if there are enough students enrolled. A list of electives is given in table 3. Students must take three electives. One is in year two and two are in year three. At least one of the elective modules in year three must be an IS elective. Note: all three electives could be IS modules. It is possible to have both year two and year three students in the same module. Electives provide the only opportunity for students to be with students not enrolled in the BAIS course.

Conclusions

This paper shows the current state of Information systems education in Hong Kong. It points out the cultural differences which underlay the Information Systems programs at City Polytechnic of Hong Kong and are representative of higher education in general in Hong Kong. It also points out that Hong Kong as a semi-sovereign entity of six million people can tailor advanced education to the needs of its society. What has not been discussed is the 18,000 students who go abroad, about 1/3 to the US, for education and the number of programs that flourish in Hong Kong where the final degree is awarded by overseas institutions (and bring outstanding financial reward to the awarding university).

The graduate of the Hong Kong programs has been trained to be of immediate productive use to Hong Kong employers. As such the programs in Hong Kong are more vocationally oriented. Programs are often accredited by professional organisations often in the United Kingdom. This leads to the replacement of the term "degree" with "qualification". Students are aware of which programs are accredited by the various professional bodies and these courses are in greater demand.

References

Bachelor of Arts (Honours) in Information Systems, Brochure, Department of Information Systems, Faculty of Business, City Polytechnic of Hong Kong, Undated

Jordan, Ernest, "The IFIP/BCS IS Curriculum as a Design Tool", Working Paper Series No WP94/02, IS Department, City Polytechnic of Hong Kong, January 1994.

Master of Arts in Information Systems, Brochure, Department of Information Systems, Faculty of Business, City Polytechnic of Hong Kong, Undated

Postgraduate Prospectus [1994 Entry], City Polytechnic of Hong Kong, 1994

Roberts, David, Ed. Hong Kong 1992, Hong Kong Government, 1992

Wong, E., Jordan, E., and Chamberlain, D. M. The IFIP/BCS Curriculum: Its Contribution to an Undergraduate Program in Hong Kong, ISTIP92, Information Systems Teaching: Improving the Practice, England, Sep, 1992.

Tips for Teaching Information Systems: Sharing Stories of Success and Failure

FACILITATORS

Eli B. Cohen
Wichita State University

Bruce White
Dakota State University

ABSTRACT

This session provides the opportunity for all participants to benefit from one another's experiences teaching Information Systems (IS). The facilitators, who are co-editors of the "Tips for Teaching IS" column of the ***Journal of Information Systems Education***, will lead the audience in developing solutions to the problem we all share — how to teach IS so that students learn, and enjoy learning. While the actual topics will come from the audience, here are some topics we anticipate will be covered:

- Teaching Programming
- Making Groups Work
- Teaching Ethics without Preaching
- Getting non-majors interested
- Teaching technical topics to non-technically prepared students

The session is designed to provide lively and animated discussion. Participants can expect to walk away with some concrete ideas on how to make their teaching even better.

EDUCATIONAL EFFECTIVENESS OF COMPUTER SOFTWARE

Netiva Caftori
Computer Science Department
Northeastern Illinois University (NEIU)
312/794-2739 UNCAFTOR@UXA.ECN.BGU.EDU

ABSTRACT

An attempt to demonstrate that not all educational software is used as intended established that predefined objectives are far from being achieved. We concluded that a balance has to be achieved between letting a child take charge of her own learning and telling her how to proceed.

The research was done at Old Orchard Junior High School (OOJH) in skokie, Illinois, during 1992-93. OOJH is the only middle school in District 68 in Skokie, and is recognized nationally for excellence. It comprises of 6th, 7th and 8th graders.

The three computer labs at OOJH

- One Apple lab is used solely for two keyboarding classes and two tool classes (spreadsheets and databases) during four periods each day.
- A second lab contains twenty-five Tandy lap-top computers which are transported to classes as needed for wordprocessing usage.
- The third, or <u>main</u>, lab contains 30 Apple computers with color monitors, some connected to printers. Besides Language Arts, music and wordprocessing, very rarely any other subject is covered there.

On the whole, classes are scheduled at 50 to 70% of the time throughout the day in the main lab. However a recent move by a faculty committee to shift the present keyboarding/word processing course from seventh or eighth grade to sixth grade has occurred. Such a move means that writing, via computers, will be significantly enhanced, and the computer lab will be busier in the future.

Software used at OOJH

In the main lab at OOJH, there is an extensive library of software, educational and non-educational. For the lunch hours (study hall period), a list is posted of approximately 40 games available to the students to choose from. This list was compiled over the years by the lab coordinator and the lab supervisor from what is available in the school, from what they are able to get from Minnesota Educational Computing Corporation (MECC), through a license agreement, and according to past usage. Apparently, only a dozen games, the most popular software programs invariably are chosen by the students who possess only a short lunch period of 20 minutes.

The most popular software programs used are

- Shifty Sam,
- Jeopardy,
- Fraction Munchers,
- Number Munchers,
- Where in the World (or the US, or in Time) is Carmen Sandiego?
- Dazzle Draw,
- Game Show,
- Word Challenge,
- Odell Lake,
- Microzines,
- Paper Plane Pilot,
- Wood Car Rally,
- Oregon Trail,
- Pizza to Go, and
- Dog Sled Ambassadors.

Other lunch study programs listed but rarely or never used are: Backyard Birds, BlueGrass Bluff, Cavity Busters, Clearwater Detective, Factory, Fossil Hunter, Hangman, Lemonade Stand, Hinky Picky, Lewis and Clark Stayed Home, LittleTown Zoo, Lunar Greenhouse, Miner's Cave, Murphy's Minerals, Perplexing Puzzles, Science Trivia, Sun and Seasons, Time Navigator, Time Navigator Around the World, Travels with Zazoom, Trivia Machine, Vacation Nation Travel, Weeds to Trees and Wooly Bounce, Treasures for Sale, Amazing Arithmetricks, Dueling Digits and Cryptoquest.

As can be seen, the students at OOJH are provided with some very good educational software programs. We will discuss the way they are being used.

Experimental subjects at OOJH

During the three lunch periods students are free to come to the main computer lab to play educational games for 20 minutes instead of going to study hall. After school, the computer club meets in the lab solely for games, not necessarily educational.

Observations of the students were conducted in this lab during the lunch hours as they were interacting with a particular computer software. A few of the students were chosen for in-depth interviews, on a voluntary basis. Each lunch period, different grade level students attended. Consistently most of the students were males. The children were allowed to communicate in an orderly manner. Noise was not tolerated. Children were unsupervised by a teacher except for the lab supervisor, whose sole function was to distribute software and manage an orderly conduct in the lab.

Analysis of the interviews and observations at OOJH

From our own observations and those of the lab supervisors, the games used during the free lunch periods are divided into the following categories:

- **Games girls like most (frequently played):** Jeopardy, Fraction Muncher, Number Muncher, Game Show, Word Challenge.
- **Games girls like less (played some times):** Shifty Sam, Carmen Sandiego, Dazzle Draw, Oregon Trail.
- **Games girls don't like (never played):** Paper Plane Pilot, Wood Car Rally, Microzines, Odell Lake.
- **Games boys like most:** Jeopardy, Oregon Trail, Carmen Sandiego, Odell Lake, Fraction Muncher.
- **Games that boys don't like:** Game Show, Word Challenge.

It was noticed that children did not want to venture with new software as their time was limited. When they came to the lab for the first time, they usually played games that their friends have recommended.

From the software documentation, many programs were designed for a classroom environment with trained teacher guidance, which was not the case during the lunch periods. The supervisor was there to answer any inquiry initiated by the students, however it was not the task of the supervisor to be fully prepared for questions that were rarely forthcoming. It took the author of this paper over an hour and a half to get to the end of Oregon trail and learn something from it. Twenty minutes are not a sufficient time to fulfill much of the intended objectives of many of the educational software programs offered.

Besides Language Arts and Music, no teacher has incorporated related software in their discipline, so the children are left to fend for themselves when it comes to computer time[1].

Let us examine each game individually:

Number and Fraction Muncher: Many children play it for the purpose of attaining the highest score. Its arcade-like quality makes it very attractive despite the fact that there is not much for a Junior High student to learn from the game. One may argue that some mathematics facts are being strengthened. That may be true for some, however the majority of the children who play this game play it because they know their facts so well that they can proceed at a high speed without much thought.

Jeopardy is liked by many children since they are challenged to give the correct answer, win against the computer and score high. The girls that play it are usually enrolled in Advanced Studies. According to the lab supervisor, average female students are not seen at this game. This may be explained by the level of difficulty, and girls' concern about being embarrassed in front of others who may look over their shoulders, as was suggested by Huff and Cooper (1987). A great deal of trivia needs to be memorized. Some self confidence is also necessary in order to attempt this game for the first time. These two items were cited as a possible cause for the lack of girls at computer labs (Krendl et al., 1989).

Where in the World is Carmen Sandiego? is played by both boys and girls. They find it challenging. Since it requires a long time for many people to obtain results, some students don't make it their first choice.

Dazzle Draw is an attractive program for artistic creation. It allows for figurative and abstract designs which cannot be done easily by hand. For example: mirror symmetries, spray paint, squiggly lines, and filling enclosed areas with a variety of patterns.

Paper Plane Pilot and **Wood Car Rally** were played by boys only. Players were not following directions and were not putting forth much thought into selecting the values for the variables involved. Children seemed to improve their aiming skills, but more by trial and error strategies than careful calculation. A precise study that records the various strategies used could possibly determine if and how much learning occurs.

Oregon Trail is a history simulation software whose goal is to introduce children to the life of wagon travellers on their way to Oregon across North America in 1848. During the simulation, children are able to visit and learn the history of forts and other landmarks on the way, and "talk" to others on the trail about their surroundings and different cultures. Children learn to manage with little money and food, raging diseases and bad water. By stopping at any of the forts they can replenish their supplies. The prices become more and more expensive as they approach the west since food and other commodities become rarer. Children must bear the full consequences of the decisions they make.

This program has much to offer: It requires making intelligent decisions based on several, but not all, facts provided. It teaches them to collect, organize and retrieve information after its interpretation. The program presents many problem solving situations such as river crossings. Every solution has implicit consequences which are accumulated to determine the end result of the game. The game is played in an interactive mode via repeating menus and a few prompting questions. Explanations are provided for almost every choice, through submenus. As one continues on the 2400-mile trail, conditions such as weather, health, pace, rations, next landmark and miles travelled from the Missouri River to Oregon, are displayed.

The more complex a program, from the point of view of the many different features and paths to choose from, the more chances there are that it may appeal to a greater audience (Wishart, 1990). Oregon Trail is fairly complex in that it has many different kinds of challenges, some of which are ignored in part by some children, but taken into consideration by others. Some of the characteristics that appeal to many children are: reaching the end of the trail, scoring higher than before, making the Top-Ten list, hunting and killing animals, making graves, and crossing rivers successfully. The designers have succeeded in that the child comes back to the program. However, the question of whether the child has learned what was intended to be learned is far from certain.

The primary learning objective of this game is to develop decision-making skills in the face of changing and sometimes unforeseen circumstances. To this end, one has to consider alternative solutions and consequences, arrive at conclusions and act accordingly. Another objective is to develop intellectual skills by learning to compare and classify things, ideas, events, and situations on the basis of similarities and differences, and group them into categories.

The richness of the subject matter and the relationships which cross disciplines (such as social studies and language arts) lends itself to many individual and group activities. A manual is provided with worksheets for students and guidelines for teachers offering a multitude of suggestions for activities. Due to a lack of time for using the software, the absence of teacher guidance, and the desire to use the software for amusement, children use the game differently from each other, based on their personality, enjoy different aspects of it, and learn only a little from it. From our observations, Oregon Trail is liked mostly by boys who enjoy shooting animals for food. Girls may like this game but for other reasons such as reaching the destination, writing epitaphs on tomb stones, or surviving the hardships on the way. To some degree the objectives of the game are missed. Attributes that were introduced to attract children to the game actually divert their attention from the objective. One such attribute is competitiveness. It is found, for example, in the challenge to reach the end of the trail as fast as possible. Children are doing so with no regard for their companions or oxen. This goal becomes so important as to neglect the health of the travellers and their lives. The shooting of animals for food was introduced for the purpose of teaching children about different animals in different terrains, as well as part of the reality lived on the trail. This seems to have become a focus of attention to many, whether justified or not. Besides eye-hand coordination, not much else is learned. Eye-hand coordination is not one of the stated objectives of this game. The lack of other attributes, such as attractive graphics, animation and sound, causes the children to skip the readings designed to teach them about the surroundings and conditions of trail travelling. If one follows Purcell's list (1993) of what motivates children to learn one may notice that at least two items are missing in this environment: One is that children must be given opportunities to collaborate with others in the problem-solving process. Oregon Trail is not built for collaborative work. Two, children need the support, guidance and encouragement of instructors who facilitate exploration. The lunch period at OOJH is not built for such support. Especially, although tolerated, communication is not encouraged and this is one important factor in learning. Here is an educational software program that is well known and recognized for its "effectiveness" and its success with children. Educators who have studied Oregon Trail are usually impressed by the well-written program. It works for them since they take the little time necessary to read all the information and gather the data to make decisions. These educators are trained in organizing their thoughts, and use their skills and other tools to retrieve information at the appropriate time. The number of criteria for these decisions is substantial, and it helps to write them down as one goes along, or to pause for recollection. Oregon Trail has become an arcade game at OOJH. Some learning may be taking place, such as fording a shallow river and floating across a deep one, or even recognizing a tune that was popular those days, but it is occasional, less important, not necessarily registered for later recall. Oregon Trail is a software program with tremendous educational potential. However, the way it is used currently at OOJH is for amusement only.

Odell Lake is a science simulation program to identify fish and their habits in Odell Lake, Oregon. It presents concrete details that allow the learner to move from the specific to the general, from the concrete to the abstract. The general principles of predator/prey relationships and food chains are to be discovered. However this goal is rarely achieved in the current surroundings. Only boys use this program as they enjoy watching the swallowing of fish by other bigger fish and also get the satisfaction of predicting the outcome of an encounter. Some play for points and best results are posted for posterity. As this lab is not part of a science class, there is no purpose for any child to learn about fish behavior or food chain. The girls' absence from this game may be explained by the possibility that the gratification that one may get from the game in the allotted 20 minutes is not one that girls enjoy. The designers of this software here again used the attribute of aggressiveness, of beating an opponent to such an extent that some boys care mostly about swallowing fish, and the sound effects accompanying it. This is an example of unfocused experience: It is not in the children's intention to learn about food chains, and therefore they do not make any effort in focusing their attention on such details.

According to the documentation, one possible way for starting this game is with no prior guidance. Therefore, unlike Oregon Trail, where teacher's guidance was recommended wholeheartedly, the children are invited to explore on their own as one possible strategy, followed by a regular class session. However such circumstances are not guaranteed, not at OOJH, and possibly not at other schools. If the goal of the software is to teach predator/prey relationship, this goal may be only accidentally achieved. A teacher should be packaged with the software at the selling point. That teacher could give a short workshop to the intended teachers on how to ideally use the software. Selling the software and only writing in the documentation how to use it is, in many cases, not enough. Very few teachers take the time and effort to read through documentation. It is easier and faster to learn by example, or at least watch someone else demonstrate, and thus get motivated to browse through the documentation.

Conclusions and recommendations:

From our observations and interviews at OOJH it was gathered that the children do not follow the software available to them as intended by the designers. The "lunch time" computer lab was set up to provide the children with some supposedly stand-alone educational software. The sincere intentions of the administration were not to provide the students with fun-like activities alone. In the equivalent study hall it is not permitted to play games either. Each child used his limited time to fulfill some need for relaxation, fun, curiosity and natural desire to learn, or simply because it was better to be at the lab and "play" than to be at the study hall and study. Each one of the children used a program that was challenging to that person, to the extent that it did not become frustrating if too hard. On the other hand, students sometimes stopped using some programs because they were "boring", too slow, not challenging enough. Each one of them was attracted at first to a particular software game because it was readily available, already known to them, recommended by a friend, or easy to learn. Each child came back to the particular game because some characteristics in it, such as scoring high, attaining a goal, graphics, or answering correctly, were appealing. These characteristics were put there by the designers to attract the children to explore further.

Much of the educational software existing today and surveyed here, requires teacher's guidance and longer time span of at least 40-50 minutes of a class period. Even the "self-explanatory" software such as Odell Lake, needs much improvement in order to render it really self-explanatory and educational at the same time. Because of the lack of such a teacher the expectations of the designers cannot be satisfied. In less than a perfect setting, not much of the intended learning can be accomplished. It is not enough to employ characteristics that render the software attractive, because these characteristics become a diversion from the real goals. The actual educational objectives cannot be met with this tactic alone. The computer technology is not used to its capacity. Even with the limited technology that exists at OOJH, software may be written especially for this kind of environment. The focus needs to be changed as the ideal situation does not always exist, not even in such a good school as OOJH, much less in poorer school districts, and certainly not in many households where parents are not there to guide their children on the computer. Even if teachers are present, they may not be able to guide each child appropriately, make sure there are no side tracks, and help bring back the straying child to the right path. They may not be 100% attentive to every child, providing the feedback necessary for good learning. A program could be designed to store all moves made in the computer memory for later retrieval. It could prompt each individual child, for example, according to her record or past behavior, and ask questions or give hints, something that even a very attentive teacher may not know, due to the teacher's limited memory and other concerns.

Unless the program takes the role of an absent teacher, by prompting questions, summarizing in appropriate phases, guiding students and keeping them on track toward the expected goal, the intended learning will not occur. Other learning may occur, even better, but not necessarily the one that the software designers planned for. The children, left on their own and wanting basically to have fun, will divert their attention to the features that appeal to them most, and will bypass many of the learning opportunities. Until designers of software are certain of what features do both, attract children and teach, they should adopt pedagogical methods that many teachers use in a hands-on environment, or when using manipulatives (e.g., remind children of what the goal is, point out to inconsistencies in the child's actions), and incorporate them into the software, avoiding at all cost the most uninteresting method, i.e., lecturing.

Educational software needs to be made more self-explanatory by incorporating some teaching techniques and behave as a teacher rather than counting on teachers being present. It needs to play the teacher's role and yet be motivating. If the computer can assume such a role, it would accomplish something that is dearly needed: It can provide specific attention to each child, immediate feedback, and individual guidance.

What we concentrated on was the simple setting of individual computers, with limited memories, in an ordinary school, and the learning that is not taking place because of the way that the software is used, but that **could** take place given an altered software or an altered environment.

References:

Huff, C. and Cooper, J.: Sex bias in educational software: The effect on designers' stereotypes on the software they design. J. Applied Soc. Psych., 17(6):519-532, 1987.

Krendl, K. A., Broihier, M. C. and Fleetwood, C.: Children and Computers: Do Sex-Related Differences Persist. Journal of Communication, 39(3): 85-93, Summer 1989.

Purcell, J.D. Designing new learning environments, Supplement to T.H.E Journal, 2-3, October 1993.

Wishart, J.: Cognitive Factors Related to User Involvement with Computers and Their Effects upon Learning from an Educational Computer Game. Computers Education, 15(1-3): 145-150, Great Britain, 1990.

Soft Computing

-the Emergence of a New Discipline

Romuald Jagielski
Department of Computer Science
Swinburne University of Technology

Email: rom@saturn.cs.swin.oz.au

Abstract

Soft computing is an emerging new discipline that combines computational methods which share similar inexact, approximate reasoning approaches in attempting to resolve complex problems. The basic components of soft computing are fuzzy techniques, artificial neural networks, evolutionary computation, and probabilistic methods. The necessity of soft computing education is discussed, and a syllabus for undergraduate study proposed.

1. Introduction

Since Kuhn's[6] discussion of the concept of scientific paradigms in 1970, "paradigm crisises" or "paradigm shifts" were suggested in various disciplines. An interesting example is soft system methodology[9]. This new system analysis approach was proposed as an alternative to conventional system engineering, which has frequently been unable to cope with the complexity and uncertainty of real situations. The rebirth of connectionism in the 1980s can be considered as another example. As a result, after a long period of unfulfilled promises, many researches started to talk about "paradigm shift" in the field of artificial intelligence.

This paper is an investigation of the shift from traditional "hard" or "crisp" thinking about computing towards a "soft "approach. The suggestion is made that this shift should adequately be reflected in information technology education. This will provide a better response to the vague, uncertain, stochastic, fuzzy and often non-deterministic world in which we live.

2.Soft computing

Up until the mid-eighties the dominant tendency was to use computers in a manner akin to mathematical proof and deductive logic (as Prof.V.R.Vemuri[10] says, we were "indoctrinated to believe that the epitomy of intellectual activity is a good mathematical proof"). However, this approach has been largely ineffective when applied to complex problems. One may say that all the simple, linear and deterministic problems have been already resolved. Those remaining unresolved are recalcitrant to classical methods and require a new science. This new science is computational science, a science which exists alongside theoretical science and experimental science - now we can compute and show, rather than derive an existence proof. [1]

Prof. L.Zadeh of the University of California proposed the name "soft computing" as a term for a combination of different techniques that he represented by a triangle. One side of this triangle was fuzzy logic, a second side probabilistic reasoning and the third learning(neural networks). Other researchers extended this model further.

Generally, soft computing consists of all the computational disciplines that can capture the inherent uncertainties of everyday life. Soft computing provides methods for solving problems which are:

- non-linear,
- non-deterministic,
- complex.

These problems may be combinatorially explosive, highly dimensional, and may have multiple objectives, subjective decision criteria or linguistic descriptors. Some examples are forecasting chaotic series such as the stock market, stabilisation and control of dynamic systems, classification of ground cover from satellite images, synthesis of drugs, etc.

3.1. The structure of soft computing

The major components of soft computing are fuzzy logic, neurocomputing, evolutionary computations, rough sets and some more classical methods of approximate reasoning.

[1] Possibly a shift paradigm can be perceived in mathematics too - recently scientists have proposed a computational proof that offers only the probability - not the certainty, and the term "video proof" has also been used [3].

3.1. Fuzzy logic

Fuzzy logic[11] was introduced by Lofti Zadeh in the 1960's as a means of modelling the uncertainty and ambiguity of natural language. Fuzzy logic is a superset of conventional two-valued logic that has been extended to handle the concept of partial truth, ie. truth values between "completely true" and "completely false". This responds well to our intuition, since we live in an analog world, not a digital one. The major notion of fuzzy reasoning is the concept of a membership function. An object may belong partially to a set. The membership function is a real number between 0 and 1, and expresses the grade of membership. This allows us to use in the process of problem solving or modelling some fuzzy terms, such as "high", "sort of", "more or less low", etc. These kind of linguistic terms are used in heuristic rules, creating, for example, a system of if-then rules with linguistic variables (fuzzy rules). Comparatively simple techniques provide approximate (or fuzzy) reasoning, ie. the inference of a possible imprecise conclusion from the set of possibly imprecise premises.

3.2. Neurocomputing

Neural networks[2] are information processing technologies inspired by studies of the brain and nervous system. They consist of a number of simple processing elements and a set of interconnections with variable weights and strengths. Information is processed by a spreading of patterns of activity in a massively parallel manner. A neural network is trained rather than programmed. It can also act as an associative memory.

Neural networks have remarkable features: they learn from examples; they can retrieve stored information from incomplete and noisy input; they are fault tolerant; they can be self-organised. That is why the past decade has seen an explosive growth in neural networks applications, which range from a jet engine diagnostic systems to stock exchange prediction, from sonar interpretation to risk assessment.

3.3. Evolutionary computing

Computer science has been unable to address several difficult problems, such as target recognition or adaptive navigation, which are routinely solved by biological systems. Genetic (or evolutionary) computing[5] is a powerful method of programming which relys on developing systems that demonstrate self-organisation and learning, in a similar, though simplified, manner to the way in which biological systems adapt.

A genetic algorithm is an interactive procedure maintaining a population of structures that are candidate solutions. During each temporal increment the structures in the current population are rated for their effectiveness, and on this basis a new population of candidate solution is formed. This process involves the use of "genetic operators" such as reproduction, crossover and mutation. The structure (solution) which emerges is a consequence of fitness.

Genetic algorithms, like neural networks, learn from experience. They find many applications in area of dynamic process control, design of engineering structures, pattern recognition, and others.

3.4. Rough sets

Rough sets[7] is a new methodology concerned with the classificatory analysis of imprecise, uncertain, or incomplete information. The key ideas behind rough sets are the approximation space and lower and upper approximations of a set. Any subset defined through its lower and upper approximations is called a rough set. By applying the rough set theory it is possible to deal with uncertainty in data, analyse hidden facts in data, create classification and prediction models, etc.

The concept of rough sets should not be confused with the idea of fuzzy logic. Rough sets are based on relationships between classes of objects whereas fuzzy sets describe intensities of objects within the same class. Applications of rough sets lay in the same broad spectrum of business, industry and science as that of neural networks.

3.5. Traditional method of approximate reasoning

A number of theories have been devised to deal with the problem of uncertainty[1]. Particularly important for expert systems is inexact reasoning involving uncertain facts, rules, or both. As well as comparatively new methods such as fuzzy logic and neural networks, techniques based on classical probability have been used. Two theories particularly have been frequently used: Bayes' theorem and the Dempster-Shafter theory.

4.Applications

Soft computing techniques provide an excellent tool for interfacing the real word of measurements, and the conceptual word embodied by the language[1], and attracts the attentions of investigators of many branches of science. The 3rd International Conference on Soft Computing was held in August 1994 in Japan[4].

Applications of soft computing, especially fuzzy logic, have created a huge industry, particularly in Japan. However, one must be remember, that soft computing methods are in the process of rapid development and constant improvement. Typically, a particular method works well on a particular problem, but not on others. Frequently it is impossible to explain the results or reproduce the same results by repeating the experiment. Often, however, only soft computing methods are capable of solving many practical problems of

medicine, military, economics, etc. Here are some examples of soft computing applications:
- predicting, for example, of the stock market behaviour,
-control, for example, of automated space vehicle control,
-classification, for example, of acute toxicity of poisons,
-image recognition,
-speech processing,
-natural language processing,
-robotics,
-optimisation, etc.

5.Soft Computing vs Artificial Intelligence

The limitations of the purely symbolic approach in artificial intelligence are well known. Soft computing is complimentary to traditional artificial intelligence. Soft computing offers an alternative way of computing, without being involve in discussions about intelligence. Obviously there is still a need to teach "traditional AI", and such important disciplines as search, learning, knowledge representation, vision, natural language processing, and possibly expert systems. The essence of soft computing is qualitative and approximate reasoning, inexact and approximate computing. Since the core of soft computing is neural networks, the closest equivalent subject/unit taught at universities might be called Neural computing. However, the term "soft computing " reflects better the current tendency in science generally, and computer science and information technology in particular.

6. Conclusion

The techniques discussed above are successive steps in the evolution of computing. They respond to the need for dealing with higher level of complexity in the uncertain and inexact world. The term soft computing defines a coherent discipline which is a natural and necessary extension of the traditional computing, and as such, requires some changes in the information technology educational system.

The course in soft computing will be run for undergraduate students in X University in 1995, according to the attached syllabus (Appendix).

References

[1] Giarranto, J., G.Riley "Expert Systems. Principle and Programming" , PWS-KENT Publ., 1989

[2] Hecht-Nelson, R. "Neurocomputing",Addison-Wesley, 1990

[3] Horgan, J. "The Death of Proof", Scientific American, October 1993, pp.74-80

[4] IZUKA'94. 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing, August 1-7,1994, Fukuoka, Japan. 1st Announcement and Call for Papers.

[5] Koza, J. "Genetic Algorithms", The MIT Press, 1993

[6] Kuhn, T.S. "The Structure of Scientific Revolutions", The University of Chicago Press, 1970

[7] Pawlak, Z. "Rough Sets. Theoretical Aspects of Reasoning about Data", Kluwer , 1991

[8] Ralescu, A. "Fuzzy Sets, Fuzzy Logic and the Goals of Artificial Intelligence" Proceedings[of The First New Zealand International Conference in Artificial Neural Networks and Expert Systems, November 1993. IEEE Computer Society Press

[9] Rosenhead, J. (ed.) "Rational Analysis for a Problematic World", John Wiley&Son, 1989

[10] V.Rao Vemuri "Problems and Issues in Neural Network Application: A Personal Perspective" Opening speech on Artificial Neural Networks and Expert Systems Conference, Dunedin, November 1993

[11]Yager, R.R., L.A.Zadeh (eds) "An Introduction to Fuzzy Logic Applications in Intelligent Systems", Kluwer, 1992

Appendix:

Soft Computing
A syllabus for undergraduate students

Aims:

To introduce a set of computational methods, such as fuzzy logic, neural networks, and genetic algorithms, which can provide techniques for the solution of complex problems.

Prerequisites: any course in advance programming

Context:

1. Introduction.

Methods of inference. Deductive logic. Induction. Approximate reasoning.

Symbolic and sub-symbolic processing.

2. Fuzzy logic.

Fuzzy sets. Fuzzy logic. The basics of fuzzy systems. Fuzzy systems applications.Case study.

3. Neurocomputing.

An introduction to parallel processing in networks. Perceptons. Multi-layer networks (back-propagation). The associative memory problem (the Hopfield model). Unsupervised competitive learning. Other neural networks

4. Rough sets.

Basic principles and foundations. Knowledge representation. Reasoning. Alications in decision making, data analysis, dissimilarity analysis. Case study.

5. Evolutionary computations

Foundations of evolutionary computation. Genetic algorithms. Applications of genetic algorithms. Case study.

Teaching Method:

Lectures: 2 hours/week
Laboratory: 1 hour/week

Assessment:

a) two assignments (fuzzy logic application and neural network application)

b) final examination

Recommended text:

Textbook - not available. Relevant books and papers will be recommended.

TEACHING COMPUTER GRAPHICS APPLICATIONS IN THE DPMA MODEL CURRICULUM

Reza Sanati
Alfred L. McKinney
Krishna K. Agarwal

Department of Computer Science
Louisiana State University in Shreveport
One University Place
Shreveport, LA 71115

ABSTRACT

Increases in computer technology have greatly enhanced graphics capability. This has fostered widespread computer graphics applications in information systems. This prompted us to modify our curriculum to include a course which would acquaint our students with these important applications. We developed a computer graphics applications course and we have taught it several times. We believe that this is a great addition to our program. This paper describes the course and the teaching methodology and we recommend that such a course be added to the DPMA model curriculum.

INTRODUCTION

We have worked closely with representatives from a local air force base on several Graphical Information System (GIS) projects. We also discussed graphics applications with members of our Tri-State Advisory Council which consists of key computer people from local industry. These projects and discussions help us realize that our students should be more knowledgeable in computer graphics and its applications.

We addressed this need by developing and teaching the course Computer Graphics Applications (see the Appendix for our syllabus). This course includes a brief history of graphics, terminology, input and output devices, output primitives and a discussion of some of the major algorithms in graphics. Then the major focus shifts to graphics application systems. These include a presentation system, a computer-aided design (CAD) system, an animation system and a Geographic Information Systems (GIS). We have used Harvard Graphics for the presentation system, IBM CAD and DigiCAD for the CAD system, Autodesk Animator for the animation system and PC-Globe as a primitive GIS.

TEACHING METHODOLOGY

This course has been presented as a one semester course for three credit hours. It usually meets for three hours one night a week. The class activity is divided into two major parts: the theoretical part and the application part. The parts will be explained in the following two subsections.

THE THEORETICAL PART

In this subsection, we describe how we introduce the theoretical aspects of computer graphics to our students. This includes computer graphics concepts, principles, applications, hardware and basic algorithms.

COMPUTER GRAPHICS APPLICATIONS

At LSUS, where the students mostly work full-time and take courses as part-time students, the students may come from different backgrounds. Most of the students have not had much computer graphics background. Therefore we demonstrate a lot of computer graphics application packages in order to create and increase the students' interest in this area. This demonstration is done using on-line computer graphics demonstrations, graphics video tapes and color graphics slides on overhead projectors.

COMPUTER GRAPHICS HARDWARE

The computer graphics students need to have a good knowledge of computer graphics hardware technology. After defining terms such as pixel, resolution and other output primitives, we discuss devices such as monochrome and color Cathode Ray Tube (CRT) and other computer graphics input/output devices. After a good introduction to hardware devices, it is logical to discuss basic algorithms.

COMPUTER GRAPHICS BASIC ALGORITHMS

If students know these computer graphics basic algorithms, they understand the computer graphics software packages better. For this purpose, we have selected a textbook for this course with a good coverage of two-dimensional and three-dimensional computer graphics basic algorithms and operations [VA93]. This coverage includes topics from drawing pixels to making complete pictures. Several algorithms such as the Bresenham line and circle algorithms [HB94] are studied and then we move on to two-dimensional and three-dimensional computer graphics. Transformation operations such as translation, scaling, rotation, reflection, shearing and clipping are also covered. Other topics such as hidden lines, hidden surfaces, shading and sweeping are covered if time permits.

THE APPLICATION PART

For the application part of this course, several computer graphics software packages are covered. During this practical period, each class time is divided into two parts: the teaching part and the hands-on computer lab part. During the lecture part, we demonstrate loading, running and using software packages. Then the students take a short break. After the break the hands-on part starts. During this period, handouts are distributed to the students and they work on the computer to practice using the software package by following instructions as listed in the handout. The students may practice individually or in small teams.

The computer graphics software packages covered during the application part of this course will be described briefly in the following subsections.

PRESENTATION SYSTEMS

The book Using Harvard Graphics by Barker and Ott [BO90] is used to cover the presentation graphics system, Harvard Graphics. This book covers the basics of graphing and provides an overview of Harvard Graphics. Then each subsequent chapter covers a different type of chart for displaying qualitative or quantitative information.

Each chapter is discussed and the drawing of that chapter's chart type is demonstrated. Then the students are assigned an exercise to work and turn in by the next class period. The book has a good set of exercises at the end of each chapter which may be assigned to students to work on. The types of charts covered in this section are: Text Charts, Organization Charts, Pie/Column Charts, Bar/Line Charts, Area Charts, and High/Low/Close Charts. We have found this book well-organized plus it has a good coverage of the materials discussed above.

COMPUTER-AIDED DESIGN (CAD)

Computer-Aided Design (CAD) is the most widely used application of computer graphics. It is very important that we expose our students to this area. We have used IBM CAD and DigiCAD in the CAD portion of the course.

DigiCAD [DM89] is a powerful 2D/3D microcomputer based CAD package developed by Digital Matrix Services. It is a user friendly menu-driven system that provides layer capability. It has strong support features that distinguish it from ordinary CAD packages. These features make DigiCAD more of a Geographical Information System package than most CAD systems.

IBM CAD [IC91] is a CAD system for drafting and designing using personal computers. Using IBM CAD, you can create two-dimensional and three-dimensional drawings. It is a menu driven package which does not require any prerequisite skill or memorization. The on-line help, tutorials and sample drawings provided by the software makes it easy for students to learn.

We demonstrate both IBM CAD and DigiCAD for the student by actually going through some drawing examples. For hands-on work, a set of projects have been given to them to select from and work out individually. The students have the choice to work with either IBM CAD or DigiCAD. They have found both packages friendly, interesting and powerful.

ANIMATION SYSTEMS

We use Autodesk Animator for our animation system. Autodesk Animator [AA89] is a professional animation program with powerful drawing, rendering and animation capabilities. It contains a desktop video program, an image-processing program, a presentation graphics program and much more. This software system is supported with a tutorial manual which contains seven tutorials and guides the users step-by-step. Included in Autodesk Animator are very impressive inbetweening and morphing capabilities. Also, using Autodesk Animator, it is possible to create thousands of full-screen colorful pictures and display them in sequences as fast as 70 per second. This sequence of fast displaying pictures produces high-quality animated images.

We usually show a few animation video tapes to help the students get a good idea of what animation is and its possible applications. Then we go through some tutorials for the students and show them how animation can be done simply. Following this, the students are assigned an animation project.

GEOGRAPHIC INFORMATION SYSTEMS

Geographic Information Systems (GIS) have the power of complicated analyses such as spatial analysis and modeling. They combine database and graphics capabilities. GIS supports the daily activities of automated mapping and facilities management and has facilitated several interesting applications for industries and government offices, for example, police department, city planning, etc. This includes those who are involved with managing utilities such as: electricity, water, sewer, gas, telecommunications, cable television and their related activities [ESRI90].

There are several GIS systems but we have limited the coverage for this topic to video tapes and a basic GIS system called PCGlobe [PCG94]. In this course, we first show some video tapes which contain information on what GIS is, its power and its applications. Then we demonstrate PCGlobe which contains information about almost all countries in the world and is updated annually. This GIS package does not allow any modification and the user can only retrieve information from its database but it can compare information about different countries very easily. An exercise related to this package is given which contains a list of questions about different countries and helps the students go through the software and learn about it.

EVALUATION OF STUDENTS IN THE COURSE

To evaluate the students, they are given: a midterm, a final exam, and several assignments and projects. The final exam is given in two parts, a take home part and an in class part. The take home part contains an animation project which the students are expected to do and list the steps required. The in class part has been a short, closed book and comprehensive exam over all covered material in the course. The assignments and projects have included: six assignments on Harvard Graphics, two on IBM CAD and DigiCAD and one on Autodesk Animator. Students have rated the number of assignments fair and reasonable. The grading weights which have been used are: 30% for the midterm exam, 30% for assignments and projects and 40% for the final exam.

STUDENT EVALUATION OF THE COURSE

At the end of each semester, the course is evaluated by the students. The students have found t course very interesting and have rated the course as excellent. They have strongly recommended this course to other students. They also have found the material coverage, exams, and assignments fair and typical of any college level class.

CONCLUSION

We strongly feel that computer graphics has emerged as one of the dominant factors in computer information systems. This means that it is very important that students become knowledgeable in this area. The course that has been discussed in this paper should serve well as a model for presenting computer graphics applications to students and we recommend that it be included in the DPMA model curriculum.

REFERENCES

[AA89] Autodesk Animator Tutorials, Autodesk, 1989.

[BO90] Donald Barker and Melvin Ott, Using Harvard Graphics For Business Presentations, Boyd and Fraser Publishing Company, 1990.

[DM89] DigiCAD Operations Manual, Version 3.00, Digital Matrix Systems, Inc., April 1989.

[ESRI90] Understanding GIS, the ARC/INFO Method, Environmental System Research Institute, Inc., 1990.

[HB94] Donald Hearn / M. Pauline Baker, Computer Graphics, Prentice- Hall, Inc., 1994.

[IC91] IBM CAD User's Guide, IBM Corporation, 1991.

[PCG94] PC Globe User's Guide, PC Globe, Inc., 1994.

[VA93] Vera B. Anand, Computer Graphics and Geometric Modeling for Engineers, Wiley, 1993.

APPENDIX

COMPUTER GRAPHICS APPLICATIONS

DESCRIPTION: Survey of computer graphics technology with emphasis on business graphics, Computer-Aided Design (CAD), Geographical Information Systems (GIS), and computer animation.

TEXTBOOK: Computer Graphics and Geometric Models for Engineering, Vera B. Anand, Wiley, 1993.

LAB BOOKS: Using Harvard Graphics, Barker and Ott, Boyd and Fraser, 1990

DigiCAD 3.00R Tutorial Manual by McKinney and Hicks, LSUS, 1991

Autodesk Animator Tutorials by Autodesk, Inc., 1989.

Software Systems: Harvard Graphics 3.0, DigiCAD 3.00R or IBMCAD, Autodesk Animator.

OBJECTIVES: To introduce the basic principles, concepts, terminology and methods of Computer Graphics and to provide experience with several major graphics packages in the areas of business, presentation, Computer-Aided Design (CAD), geographical information systems (GIS), and animation. A project will be assigned for each of the packages.

TOPICS:

1. Introduction to Computer Graphics and Its Applications
2. Graphics Systems
   - A. Hardware
   - B. Software
     - (1) Fundamental Algorithms
     - (2) Overview of Standard Packages
     - (3) Design of User Interface
3. Business and Presentation Graphics
4. Computer-Aided Design (CAD)
5. Animation Graphics
6. Geographical Information Systems (GIS)

A CROSS-CULTURAL ANALYSIS OF STUDENTS' PERCEPTIONS OF E-MAIL PRIVACY

Marilyn Wilkins, Professor
Lumpkin College of Business and Applied Sciences
Eastern Illinois University
Charleston, IL 61920
217-581-3726 cfmlw@ux1.cts.eiu.edu

ABSTRACT

This paper reports perceptions of American and Irish University students toward electronic mail privacy. While Irish students appeared to have a better understanding than American students regarding electronic technology and the possibilities of e-mail messages being accessed, monitored, and stored, both groups agreed that e-mail messages should be private.

INTRODUCTION

Use of electronic mail has mushroomed over the past few years. There were an estimated 19 million users in 1992 (Rothfeder). The expected growth rate is 20 to 35 percent annually (Dorath) and by 1998 the estimated number of users will be over 100 million. One of the major issues is privacy of electronic mail messages. Questions of what is legal, what is ethical, and what is practiced abound. Who owns electronic messages? What are employees privacy expectations? Who can access e-mail messages? Do organizations have the right to read e-mail for any reason? How long should e-mail messages be retained? These questions often create conflict between an organizations' ownership rights and the employees' privacy rights. In the United States, few laws govern the privacy issues of electronic mail and several legal cases between employees (or former employees) and employers are pending in the courts. In Ireland, legislation concerning privacy and control of personal data is under consideration. The threat to privacy ranks as one of the highest concerns felt by the general Irish public about computers in general (Munnelly).

Experts recommend that companies develop policies which cover these questions but few recommend specific policies. However, recommended policies are often contradictory because of divergent expectations between employers and employees. Employees are often unaware of the accessibility of electronic mail by employers. The Harvard Law Review states, "If employees knew that their electronic mail messages were completely accessible to employers, employees would approach electronic mail in a very different spirit." With growing numbers of faculty and students using electronic mail, these same questions concern universities as well as business organizations. If policies governing e-mail are to be effective, organizations should consider privacy expectations of their employees when developing those policies. This paper reports the results of a survey of Irish and American university students. The purpose of this study was exploratory and designed to determine perceptions of U.S. and Irish students towards electronic mail privacy.

METHOD

A questionnaire, consisting of 15 items related to electronic mail privacy was developed and pretested using a class of MIS students in an medium-sized university in the U.S. The questionnaire also contained several demographic questions. The questionnaire was revised after correlation analysis of the questions. The questionnaire was administered to two management information systems classes--one in the Republic of Ireland and one in the United States. Students in the classes were predominantly upper division business students with a median age of 21 years. Students at this level could reasonably be expected to have considered ethical issues in information systems. The content of the courses was similar in nature and students had learned to use e-mail in this course. The average time U.S. students had been using electronic mail was 3.3 months and the average time Irish students had been using electronic mail was 5 months.

RESULTS

About half the Irish students and a third of the U.S. students used electronic mail for personal use, with percentage of personal mail ranging from 10 to 100 percent. When asked if their school had an electronic mail privacy policy, 90 percent of the Irish students, and 87 percent of the U.S. students said they did not know.

Perceptions of e-mail privacy was measured with a 5-point Likert type scale ranging from strongly disagree (5) to strongly agree (1). The means for the Irish students and the U.S. students are shown in Table 1.

TABLE 1
COMPARISONS OF MEANS OF PRIVACY QUESTIONS

QUESTIONS	USA MEAN (n=30)	IRISH MEAN (n=21)
1. E-mail messages stored on the computer can be accessed by network and computer administrators.	3.03	1.50
2. E-mail messages stored on the computer can be monitored by network and computer administrators.	2.97	1.86
3. When I delete E-mail messages, they are erased from the computer.	2.60	2.86
4. When I delete E-mail messages from the computer, they may not be erased from archived storage media.	3.03	2.71
5. My password implies that E-mail is private.	2.43	3.57
6. E-mail messages and files should be private.	1.93	1.52
7. E-mail is more private than telephone calls.	3.17	3.0
8. E-mail is more private than letters or memos within the firm.	3.03	2.76
9. E-mail belongs to the firm, managers have a right to monitor messages and files.	3.63	3.76
10. It is ethical to use e-mail for personal reasons.	2.30	2.62
11. It is ethical to read another's e-mail for routine sharing of information.	3.97	3.52
12. It is ethical to access another's e-mail for pressing business needs.	4.1	3.71
13. It is ethical to access another's e-mail when theft of trade secrets or computer crime are suspected.	3.67	2.71
14. It is ethical to access another's e-mail for "electronic fishing expeditions" (snooping).	4.17	4.67

The first four questions relate to perceptions students have about electronic mail technology. Irish students appear to have a better understanding than U.S. students regarding electronic technology and the possibilities of e-mail messages being accessed, monitored, and stored on archival media.

The remaining questions relate to perceptions of privacy and use of electronic mail. Both Irish and U.S. students agree that e-mail messages and files should be private and that it is ethical to use e-mail for personal reasons. However, both groups are undecided about whether e-mail is more private than telephone calls or memos. U.S. students disagreed that a password implied that e-mail was private but Irish students tended to agree that a password implied privacy.

Both groups tended to disagree that managers have a right to monitor messages and files, and that others' e-mail can be accessed for routine sharing of information. However, both groups also disagreed that it is ethical to access another's e-mail for pressing business needs, routine sharing of information, and for snooping. However, Irish students tended to agree while U.S. students tend to disagree that it is ethical to access another's e-mail when theft of trade secrets or computer crime is suspected.

CONCLUSIONS AND IMPLICATIONS

Privacy issues related to the use of electronic mail will continue to be important concerns for businesses and universities as electronic mail continues to grow by leaps and bounds. While both universities have electronic mail policies, students were generally not aware of any policies. In both instances personal use is not forbidden nor is privacy assured. In both cases, the policy (unwritten) is that electronic mail is not monitored. While the sample size is small, the students in MIS classes should be more knowledgeable than the typical university student concerning electronic communications. Both universities could do better jobs of conveying the policies to users. Organizations would do well to ascertain attitudes of employees toward electronic mail privacy issues before setting e-mail policies. Privacy issues for users of electronic mail should be an important topic of discussion in MIS classes.

REFERENCES

Addressing the new hazards of the high technology workplace, Harvard Law Review, (1991), 104(8) pp. 1898-1916.

Donath, B. (1993, April 16) E-mail emerging as communication medium of choice, Marketing News, 27(9), p. 13.

Munnelly, B. (1987, December) Whose file is it anyway? Irish Computer, pp. 19-20.

Rothfeder, J. (1992, September) E-mail snooping, Corporate Computing, 1(3), pp. 168-174.

Downsizing of Information Systems...Hardware, Software and Budget Considerations

by
Dr. Karen A. Forcht
Dr. Joan K. Pierson
Department of Information and Decision Sciences
College of Business
Zane Showker Hall
Harrisonburg, VA 22807
703-568-3057/3064 (O)
703-433-2545 (H)

ABSTRACT

Many corporations, in these tough economic times, are downsizing their over-all operations. A fundamental shift today is to assess how information is produced and used throughout the organization. Some of the key elements to be addressed in this downsizing effort are the software currently in use, hardware constraints, and budget considerations. All application areas and user needs must be carefully studied and analyzed before the downsizing process begins in order to produce an end result that will meet the organization's information needs.

INTRODUCTION

A fundamental shift is now occurring in the way businesses go about using information systems for maximum efficiency and effectiveness.

The key issues in the tight economy of the 1990's when evaluating information resources are:

1. price and performance
2. productivity
3. scalability

<u>Downsizing</u> is the process that organizations must undertake to get from the current generation of information technology (IT) to the next generation. (2)

Some of the areas that are considered for the downsizing process are:

- data entry and editing
- statistical and mathematical analysis
- decision support
- highly interactive applications (i.e. computer-aided design)

Applications that generally should not be considered for downsizing due to their criticality include:

- mission critical applications
- those with high security requirements
- high volume data base updating
- complex applications that span organizational boundaries

Another term that is often used in place of downsizing is rightsizing, which involves downsizing by selecting the right, or appropriate, platform for each part. (2)

SOFTWARE CONSIDERATIONS

When it comes to downsizing software to work on new hardware platforms, there are generally three main choices:

- buying new software
- reengineering current software
- downsizing current software

New Software

The trend of downsizing away from the mainframe has brought about new software for the mainframe that makes it more useful. The first new type of applications are called software distribution management products. These applications allow electronic distribution of upgrades and new applications to all connected work-stations. The software also allows packages to run on work-stations instead of the mainframe, so mainframe overhead can be reduced. A final feature of the new application is easy transferral of data into standard desktop applications such as word-processing and spreadsheets. However, this software does not allow current mainframe applications to be downloaded onto PCs. The cost of the software is approximately $30,000 for the mainframe component and an additional $200 to $300 per work-station.

Other related mainframe applications include on-line report viewing products, outboard console automation systems, and capacity and performance analysis tools. These types of applications give companies the ability to more closely monitor the mainframe's performance. For example, the applications can keep track of which software is being use and by which work-station. This allows management to know which software is used the most and which in not, allowing for storage of only the most needed applications for the future. This will leak to reduce software costs. (5)

Reengineering Current Software

The second option is to reengineer current software and add features that the company needs. This particular process works best for large corporations with a backlist of aging applications that need to be upgraded. When reengineering software, three steps are involved; the first is reverse engineering. During this step, the existing application is broken down into the most basic form that can be revised and updated. The second step is revising the specification model. This step includes adding new capabilities, removal of obsolete capabilities, and removal of redundant code. The last step is forward engineering. During forward engineering, computer-aided software engineering (CASE) is used to create the new code for the application. (5)

Downsizing Current Software

The final option with software is to downsize the current applications so that they can be used on PCs. This option works best with older applications that are not designed for cross-platform use. Some examples of these applications are departmental applications, query applications, packages that are being enhanced anyway, and applications that scale well. A bad choice of software to downsize is andy application that relies on the mainframe infrastructure as the conversion costs of the software will be greater than the cost savings of the new hardware platform. When downsizing software from mainframes to PCs, a few problem areas exist: The database portion of the application, terminal interfaces (mainframes use transaction monitors while PCs use graphical user interfaces), application generators, debugging tools, and expert system shells. (9)

HARDWARE CONFIGURATIONS

Most downsizing projects result in employing smaller classifications of computer hardware and more sophisticated communication or networking schemes. The only types of central processing units available when most organization initially adopted computers were mainframes and minicomputers. Since then, technology has changed drastically, however, IS operations have not always taken full advantage of these changes. Mainframes and minicomputers, in general, are not being eliminated by downsizing but their use is being limited. Many industry experts believe that mainframes will continue primarily as a warehouse for complex applications, systems management applications, security programs, and data storage. Microcomputers have advanced drastically in MIPS and clock speed, but they still cannot compete with minicomputers and mainframes in I/O capacity and data storage. The 486, Pentium, and P6 processors can all exceed the MIPS and RAM of lower-level mainframes of the recent past. Larger corporations generally are unable to migrate to PCs and PC networks, however, because of the data storage and I/O issues.

Consolidation of IS

Organizations undergoing downsizing of IS operations are consolidating computer centers, networking work-stations and PCs, and de-emphasizing traditional mainframes and minicomputers. Important factors in the decision-making process are the size of the organization, data storage requirements, throughput or input/output (I/O) requirements, CPU or processing speed, software, data distribution, financial concerns, personnel, and organizational plans. Furthermore, companies must be aware of their preferred mode of communication, response time, security issues, and acceptable service levels.

Technology Development

Complication the decisions that IS managers must make is the rapid development of technology. Reduced instruction set computing (RISC) chip architecture has burst onto the scene as a valuable component to downsizing projects. The RISC philosophy is that computer speed is maximized when all but essential functional instructions are housed in the software. (6) Recently developed multiprocessor systems are competing as mainframe alternatives. These machines have one to twenty-four off-the-shelf microprocessors; RISC chips, 100 or more million instructions per second (MIPS) processing speed, and they support thousands of users. These systems do not yet compete with upper-end mainframes in disk storage and I/O capacity, however. (7) Minicomputer and mainframe manufacturers, not to be outdone, are marketing dual-processing systems to improve processing speed and flexibility. For example, the Unisys A Series mainframes provide a mainframe processor as well as an Intel 486 CPU to permit applications under MCP/AS and UNIX or OS/2. These systems are expected to replace the gateway and/or servers in networks. (8) Smaller companies running simplified applications may be able to employ "superservers", or enhanced microcomputers, which are smaller multiprocessor systems. Regardless of the hardware system, PCs are steadily replacing dumb terminals to provide more and more end-user computing capabilities, and these PCs are being connected in Local Area Networks (LANs) and Wide Area Networks (WANs).

Hardware and Software Policies

Policies must be established for the downsizing project to be successful. One of the reasons policies are so important is that they will help maintain information technology (IT) compatibility. Policies will assure that an acceptable level of systems conformity occurs between all the various departments of an organization (i.e. Production, Marketing, Accounting, Finance and upper management). By having established buying procedures, coupled with an approval process, the company can guarantee that hardware and software disparity is minimized.

Hardware and software disparity occurs when many products from many different vendors coexist in an organization. Because a standard operating system does not exist, a lack on connectivity occurs when an interface of proprietary systems (including hardware and software) from different suppliers is attempted. The following is an example of disparity. A 3270 IBM terminal cannot be successfully attached to a Digital VAX minicomputer without the purchase of additional software and hardware. The extra hardware and software purchases make the connection possible but are only needed because disparity exists between the IBM and Digital products.

Isolated buying or development of applications is one of the main causes of software disparity resulting from each department developing or purchasing software for use only in that department. Department managers do not consider the effect of such actions on the organization as a whole. When this disparity occurs, the information flow is crippled because data cannot be shared between the various departments without the use of closely connecting hardware and software.

With properly established policies, problems such as the ones discussed previously will not occur. If the organization makes departments obtain approval of hardware and software acquisitions, the possibility of a disparity occurrence is much less likely. Approval could be conducted through centralized management or by a buying agent of the department who would be held responsible for all purchases. Another added benefit of established policies is a decrease in systems costs. The cost of hardware an software will be minimized because there will be no need to purchase the expensive connecting hardware and software associated with disparity.

Inventory of Current Hardware and Software

A company must analyze its current system before it begins a downsizing project. One of the key factor in systems analysis is an inventory of the current hardware and software resources of the firm. The analyst must know the current system (hardware and software) and how the system meets and does not meet the needs of users. There should be documentation of the current system's hardware and software. If documentation cannot be found, information could be obtained by questioning departmental personnel familiar with the system. After the analyst performs the fact-finding search and has sufficient information on the existing hardware and software, then he or she can look into ways in which the existing equipment can be used in the new downsized system. Also the inventory will let the analyst know which departments ore using what hardware and software. Not only will the inventory assist the analyst in maintaining current hardware and software, and thus cut costs, it will also aid in deciding on which parts of the system are going to have to be modified or eliminated to achieve the downsizing objective. The detailed and precise inventory will provide a stable foundation on which the downsizing project can begin.

Capacity Requirements

One of the crucial planning factors of downsizing is capacity requirement projections for the system. The hardware and software must be able to meet the current and near future capacity needs of the organization. Company strategy and scope management must be considered. The future capacity needs of a firm with a growth strategy would be greater than those of a firm with a status quo strategy. Also, when looking at capacity requirements, the market environment of the company must be researched. A dynamic market could mean that significant fluctuations in capacity needs could occur, therefore, the system would have to be able to adapt to the market changes. Hardware upgrade-ability would be very important. Another consideration would have to be the different processing needs of the organization. Transaction processing usually involves large spreadsheets and number-crunching processes. Because transaction processing involves number-crunching and large amounts of data, the capacity requirements are very different for decision support systems. Transaction processing, because of the large amounts of data, usually requires the use of a mainframe or minicomputer. Decision support processing can often be accomplished on a PC-based LAN system because the amount of data involved is much smaller.

BUDGET CONSTRAINTS

Many companies are streamlining their operations and paying more attention to financial concerns. Downsizing is the word used to describe the way applications software is made to perform the same or more functions on a smaller class of computer or networked system.

First, it is important to determine the companies objectives, goals, and reasons for downsizing. Because downsizing objectives can be wide ranging it does not necessarily always relate to saving money. The initial investment or budget can be significantly different depending on the companies objectives and reasons for downsizing. A firm that considers downsizing for short-term financial relief would require a different approach than a firm that wants to position its product or service for the future. Exhibit 1 lists objectives and possible courses of actions as well as the costs associated with each alternative.

The next step is to estimate the costs of operating and maintaining the existing mainframe and software. At this step the systems analyst must make sure the required software is available for the downsized-hardware platform or that it can at least be developed within a reasonable time at an acceptable cost (1).

The third step is to determine the costs of restructuring the information system architecture. At this point, costs, other than hardware, may be quite high. Development, support, and testing time can represent substantial costs when making the transition to a new system. Training costs and the costs associated with trial and error to make PC and mainframe applications integrate properly are often seriously under-budgeted, most often by more than 100% (1).

The final step in the cost analysis is to compare the two costs and assess the advantages and disadvantages of the downsizing proposal. An often neglected idea to save money in the information system department is to take large, technical applications off the corporate mainframe and put them on a local area network. A relatively small group of users on the LAN can increase response time, thereby reducing MIPS, the measure of processing power, and increase overall capacity on the corporate mainframe. The increased capacity on the corporate mainframe will free storage space for greater utilized financial and accounting packages. Once the LAN is operational, lower mainframe upgrade costs will also be realized.

Not all costs are directly measurable. When considering a downsized technical environment, more than just cost savings need to be considered. Management should look at the opportunities to align business strategies with technical strategies. Management can now give the power to the user to perform an array of tasks that, until this point, had not been possible on the old system.

Exhibit 2 lists a possible budget for a major gas company that wanted to move to a LAN-based network with distributed personal computers for its tow subsidiaries. The capital budget is primarily used for computer hardware and large-scale components of the information system. The operating budget is typically used for software, conversion, and training cost associated with the downsizing project. No matter how complicated the downsizing project, cost-benefit analysis summarized the relevant costs and benefits in detail to provide management with enough detail to decide whether to proceed with the project. Hiring outside consultants is also popular, given the complexity of computer systems in larger firms and lack of qualified personnel in smaller companies.

Whatever the reasons, downsizing is happening at an increasing rate, and a company's objective will ultimately determine the budget and financing alternatives.

<u>Security</u>

Companies downsizing from mainframes to distributed systems are beginning to realize that security is an important issue. Who will have access to the data? This question is being asked by all MIS personnel who have become concerned that confidentiality and control are becoming difficult to maintain with the now or proposed downsized system. While most companies have installed some type of anti-virus protection on their computer system, security packages, which include access control, authentication, encryption, etc., have not yet been established within the downsized system. Because many of the downsized applications are moving to LAN-based systems in which power is given directly to many users via the PC, accessibility from insiders and outsiders has raised questions among system analysts. The companies' main concern in establishing a new security system is to ensure data control and increase protection from unauthorized access.

CONCLUSION

Downsizing efforts cause a great deal of time, effort, and money to be spent. Coupled with the mental stress of employees dealing with this dramatic change, organizations need to plan and administer their efforts carefully in order to ensure a workable, productive outcome. Companies who view downsizing as an opportunity to offer streamlined IT services to a greater number of users will ultimately prosper. Those organizations not approaching downsizing as a chance to do a better job of IT are surely doomed to be left behind.

REFERENCES

1. Dagenais, Tom. "One Size Does Not Fit All." Information Technology, 8 August 1991, pp. 30-31.

2. Forcht, Karen A., Management Information Systems ...A Casebook, Fort Worth, Texas: Dryden Press, pp. 101-103.

3. Fosdick, Howard. "How to Avoid the Pitfalls of Downsizing." Datamation, 1 May 1992, pp. 77-79.

4. Guengerich, Steven. Downsizing Information Systems: Proven Method For Reengineering Complex Information Systems. (Carmel, Indiana: Sams Publishing, 1992), pp. 87-240.

5. McCabe, Thomas. "Tips on Reengineering Redundant Software." Datamation, 15 April 1992, pp. 71-74.

6. Milutinovic, Veljko M. High Level Language Computer Architecture. (USA: Computer Science Press, Inc., 1989), pp. 3-4.

7. Moad, Jeff. "Mainframe Power for 80% Less." Datamation, 1 Jan. 1993, pp. 67-72.

8. Musich, Paula. "Unisys continues downsizing push." PC Week, 12 April 1993, pp. 47-48.

9. Taber, Mark. "Systems Management Beyond the Glass House." Datamation, 15 March 1992, pp. 63-67.

THE CHANGING REQUIREMENTS FOR INFORMATION SYSTEMS PERSONNEL: THE IMPORTANCE OF BEING POLITICALLY ASTUTE

Karen Ketler
Computer and Operations Management
Eastern Illinois University
Charleston, Illinois 61920
(217) 581-6906 cfkjk@eiu.edu

John Walstrom
Computer and Operations Management
Eastern Illinois University
Charleston, Illinois 61920
(217) 581-5949 cfjxw@eiu.edu

ABSTRACT

Business practices in the 1990s are characterized by strategic alliances with other organizations. However, while organizations are seeking improved communications with these organization, there are internal "turf" battles. Today, systems analysts must manage internal conflict in order to develop corporate-wide applications. However, few IS personnel have received training in conflict management. Suggestions for integrating political awareness in the IS curriculum are offered.

THE BUSINESS AND TECHNOLOGICAL ENVIRONMENT OF THE 1990'S

Businessmen in the 1990s are faced with an accelerating rate of change in the external environment precipitated by an accelerating rate of growth in the use of technology. A bewildering, and often times frustrating, thought is "This is as slow as it will ever get". Future changes will occur at an even more rapid rate. In an effort to manage the rapid change, organizations are turning to strategic alliances with their customers, suppliers and other organizations. Technology is the key to many of these alliances.

The first illustration of a strategic alliance is outsourcing which is the transferring of part or all of the IS function to an outside vendor. This allows the company to concentrate on its primary business. Those opposed to outsourcing argue that once the external vendor has control of the data and the IS function the livelihood of an organization is at stake. Outsourcing is an issue of risk versus control. Ultimately, the effectiveness of the outsourcing venture rests on the strength of the strategic alliance between the customer and the vendor providing the service.

Electronic Data Interchange (EDI) provides a second illustration of a strategic alliance. With EDI, customers and suppliers can communicate (through a third party) with each other's computer systems to automate the purchasing and billing functions As a result, organizations using EDI have more effective communication with customers and suppliers, which results in improved delivery response. Organizations also report their great satisfaction with the reduced paperwork.

However, on the negative side, there have been reports of small businesses feeling pressure to join in these alliances in order to maintain their customers and suppliers. One small organization reported coercion to implement EDI if it wished to retain one of its highly valued customers.

For a small organization, the implementation of EDI including the contracting with a 'clearing house' for the routing of the data can be costly. Most of these vendors charge a monthly fee in addition to a per transaction charge. This small organization anticipated it would communicate with only one customer via EDI, and would average less than ten transactions per week. The overall cost of EDI is prohibitive for such limited use. Nevertheless the organization is faced with either losing a customer, or increasing its costs.

[This example, and all subsequent examples were obtained as a result of a 1993 study of information systems internships.]

EXAMPLE 1

Small Manufacturer of Agricultural Products

The information in this example was furnished by a faculty member who has been doing consulting work at the corporation for the past few years. Recently, the work has taken on an information systems focus.

The corporation is a small (9-employee) manufacturer of agricultural products. The organization has conducted international business with Canada, Central and South America, and Europe for a number of years. In 1993, the corporation adapted its cattle products for use with ostriches in Australia.

Until 1992, the organization was without a computer system. During the spring of that year an IBM compatible 386 (20 megahertz) system with two megabytes of RAM and forty megabyte hard drive was purchased.

The catalyst for the purchase was the realization that the organization might be faced with utilizing electronic data interchanged (EDI) in order to maintain a large and valued customer.

But, before EDI can be implemented, the staff needs to increase its computer literacy. Currently, the staff is utilizing word processing, databases and invoicing programs. Additional applications, including accounting is planned for 1994.

The organization has basically "outsourced" all responsibility for information systems to their consultant. The integration of the computer into all aspects of this business is a very slow process, due to the resistance to change.

A third illustration of a strategic alliance is disaster recovery. Two or more firms may form a partnership for IS services
in event of a disaster such as an earthquake, tornado or flood. Such an alliance saved many organizations after the bombing of the World Trade Center. These organizations had previously contracted with a vendor who operated a "hot site", which is a "backup" computer center. With only backup copies of their databases and crucial software that had been stored off-site, these World Trade Center businesses where in operation at one of the "hot sites" within a few hours after the bombing.

In addition to the growing partnership among businesses, there is also evidence of partnerships between departments within a business. In the 1990s, the use of business teams to solve corporate problems is gaining popularity. Previously, some organizations allowed each unit to optimize their departmental efficiency and effectiveness; however, many of these organizations were operating at a suboptimal level from a corporate viewpoint. In the 1990s, more organizations are stressing the corporate, rather than departmental, viewpoint.

Business teams are formed with representatives from throughout the organization. These teams are then charged with the task of solving a corporate problem. Once the problem is solved, the team is disbanded. One such corporate problem that might effectively use the team approach is the large backlog of IS application development requests.

In the 1990s, organizations must be responsive to environmental and customer demands. The huge backlog of IS application requests hinder the reaction time of the organization. Organizations are attempting to solve this dilemma by hiring more highly trained users and purchasing higher level software packages. User friendly software, combined with a more computer literate work force, has been a catalyst for the movement of the development function from centralized IS into the user department. This trend provides the stimulus for many political battles between user departments and the IS department.

Two examples of user departments that assumed responsibility for the systems development function were noted in the internship study. In the next example, the transferring of the application development effort to the user department was done with the approval of the IS department. However, the IS department offered no assistance in training the intern who would be working on the project.

EXAMPLE 2
Planning Office of a State Government Office

In this organization, a computer intern was hired to write programs in the fourth generation language (4GL), FOCUS. The intern was hired, housed and evaluated by the planning and budget office.

The intern reported that she wrote special request programs concerning employees and the organization's customers for outside agencies. Although many of these requests were optional, the office attempted to respond to as many requests as possible. In addition, the intern completed internal reports for the organization. Her supervisor verified her work for accuracy before distributing any statistics or reports.

She indicated that upon employment she received a manual and a video and taught herself FOCUS. She credits her technical coursework in providing her the drills and confidence to complete the task. She said that she has been introduced to the IS staff, and that they had offered to assist her. However, the intern indicated that she was hesitant to call or impose upon the IS staff.

In the example described next, the hiring of an intern to perform activities previously assigned to the IS department initiated a political battle.

EXAMPLE 3
Information Systems Department of a Governmental Agency

In this organization, the intern was hired and employed by the user department, but was physically housed in the information systems department.

The intern reported that the first few weeks went smoothly. He spent the majority of this time in the user department learning the needs of the department. He spent very little time in Information Systems, and no one bothered or befriended him in that department. Slowly, the amount of time he spent in the user department decreased and the time in the IS department increased.

There was resentment by the IS staff because a nonmember was "adjunct" in their department. At the directive of IS management, IS professionals were told to answer the intern's questions, but offer no advice. He reported friction and political unrest. The intern felt the "turf battles" in which he was only a pawn.

The intern was a top student, and the quality of his questions impressed the IS professionals and management. The student credited his success to his ability to work effectively in a negative political climate due to his prior knowledge of conflicts. This IS department even acknowledged learning a few new ideas from the intern.

By the end of the summer, the project was completed and documented to the satisfaction of both the IS and user departments. It turned out to be a beneficial experience. In fact, the intern was invited to apply for a permanent job in the IS department.

The outcome in this example was positive but could have been entirely different. And, a question is raised: "Who is responsible for maintaining or enhancing the project?" The user department apparently does not have the time or expertise now that the intern is gone. If the maintenance task is transferred to the IS department, the user department has successfully bypassed the prioritization and scheduling of IS tasks by its management.

Battles are not restricted to just the IS department and the user departments. Political wars are waged between user departments themselves when the workload shifts as a result of a new application development. Such a battle is noted in the fourth example.

EXAMPLE 4
Political Battles Between User Departments

A small manufacturing organization was automating the accounting function. Previously, the packing slips and invoices were typed on typewriters by individuals in two different departments.

As a result of the new computer software, the packing department would be responsible for entering the all information (which was previously typed) into a computer. Shipping charges and inventory statistics would be updated, and the packing slips printed.

From this information, invoices will be produced with only one computer command. As a result of this automation, the packing department felt that they were being given additional responsibility and work with no increase in staffing or in pay. The accounting department was also upset with the change. They feared the loss of a position.

The project was placed on hold until the fears and complaints could be resolved.

LEARNING FROM PAST MISTAKES

Analysis of the short, but dynamic, history of IS reveals a common pattern of growth in the acceptance of new technologies. This pattern was first noted in 1974 when Nolan and Gibson identified four stages of IS growth:

1. Initiation - In this stage, the computer technology is introduced in a narrow or limited focus. After some initial problems, the advantages of the technology are experienced. This leads to increased interest and a wider application of the technology.

2. Contagion - This stage is characterized by the proliferation or spread of the technology throughout the organization. Inefficiency results as each user experiments with the technology. There is little managerial guidance to prevent each user from "reinventing the wheel".

3. Control - In this stage, management in an attempt to control costs and reduce inefficiencies, establishes policies and procedures. As a result, the growth rate of the new technology is slowed.

4. Mature Use - This stage is characterized by the efficient use of the technology in the organization. Lucas (1989) notes that only a few organizations reach this stage. At this point, the technology is integrated efficiently into the management process.

In 1979, Nolan added two stages (between control and mature use) to elaborate on the IS growth controlled by management. These stages include: 1) the *integration* of the technology into different applications, and 2) the *data* administration phase which focuses on the sharing of data throughout the organization.

Although the stage theory was originally applied to centralized computing, many of the same issues are applicable in the acceptance of distributed processing. The proliferation of distributed processing has caused inefficiencies. In addition, as information systems evolve from centralized mainframe environment to distributed systems, the role of both the IS professional and the user has changed. Political battles over "turf" are common. Organizations need to develop new policies and procedures in order to reduce organizational conflict.

The role of the system analyst in the 1990s is changing. IS personnel must be politically astute in order to not only solve "turf" battles directly affecting the IS departments, but also to reduce the conflict between user departments. As the role of the system analyst changes, the training associated with the position should also be adapted to include additional emphasis on communication skills, conflict resolution, and management techniques.

The next section of this paper will discuss some of the politics of reorganizing the IS function. Then, in the fourth section, a technological solution to the political problems will be addressed. An example of how a political problem was developed and resolved in a systems development course is included.

THE POLITICS OF INFORMATION SYSTEMS

Information is power (Santosus, 1993). It is the source of many "turf battles". As technology grows, departments see their responsibility and staffs ebbing away. Many middle and lower level managers, fearing the loss of their jobs, seek security through empire building. In addition, many employees have insecurities associated with changes brought on by technology. For example, electronic data interchange (EDI) forces changes in many of the procedures and tasks of the purchasing and accounting offices.

It is also important to note that many IS employees have similar fears. First, the MIS department may lose power as users develop their own applications with fourth generation languages (4GLs) and systems shells. Second, many of the skills of the IS professionals are becoming outdated with the installation of integrated case (I-CASE) technologies and reverse engineering. In addition, many organizations are looking toward the possibility of transferring part or all of their IS functions to outsourcing vendors.

Others are laying off IS personnel and other employees in a "right-sizing" effort. As individuals fear for their jobs and as information is distributed throughout the organization, "empire building" and "turf battles" are becoming common. The war is on. Who will manage and control the organization's information . . . the user department or the IS department?

It is unfortunate that many IS managers and employers have little expertise in the political arena. Many of these individuals came up through the ranks using the older technologies, with which there was little need for interpersonal skills.

It is recommended that the IS professionals (and students) hone up on interpersonal skills. Additional information about the impact of personality types in political battles can only aid the IS professional.

Eight management strategies for resolving conflict are offered by Daft (1986). Each of these strategies is geared to either modify the behavior or the attitude of group members. A change in behavior or attitude can be brought about through:

1. Bureaucratic authority - the invoking of rules, regulations and formal authority to resolve the conflict.

2. Limited interaction - the reducing of the communication and involvement between the conflicting groups. This strategy does not solve the problem, but attempts to avoid the problem.

3. Integration entities - the use of liaisons between the two conflicting groups.

4. Confrontation and negotiations - the resolving of the differences through direct contact with the conflicting party.

5. Third party consultants - the involvement of an unbiased consultant to meet with both groups and to reestablish positive communication lines between the groups.

6. Member rotation- the rotation of members from one department to another. This allows members of the conflicting groups to understand the values, attitudes, problems and goals of the other department.

7. Supercordinate goals - rather than emphasizing the departmental goals, a corporate goal involving both departments is stressed.

8. Intergroup training - the modification of employees' attitudes through additional workshop training.

These suggestions are derived predominantly from concepts in interpersonal behavior and conflict management. They bring to mind the parable of the shoemaker's children. The shoemaker was so busy making shoes for all the other people that he did not have time to make shoes for his own children. IS departments are so busy making application systems for the other departments, they forget to make or use the systems for themselves.

Why not use IS technology to solve the disputes? As indicated earlier, business teams are growing in popularity in the 1990s. These teams, formed from individuals throughout the organization, are charged with the task of resolving a problem or dispute. Once the problem is solved, the group is disbanded, and the members join another group with a different set of issues.

Such a group can be formed to solve political battles. If technology is a catalyst for some of the problems, maybe technology can also be a solution.

A basic method for resolving disputes is the use of a group decision support system (GDSS). GDSSs include the hardware, software, people, databases and procedures needed to provide effective support in a group decision making setting. According to Stair (1992), with a GDSS, negative behavior is suppressed while positive behavior is supported.

Some GDSSs use the delphi approach which allows effective feedback and fosters innovative thinking and creativity in decision making. Other GDSSs promote brainstorming and free thinking. Another technique, the nominal group technique, allows each decision maker to participate and receive feedback. Eventually, each member must vote. The votes are tabulated, and the decision is reached (Stair 1992). Various mathematical models, such as Saaty's Hierarchical Process, refine some of the "one person - one vote" assumptions.

In the final section of this paper, a technique for developing the political awareness of future systems analysts is presented. The implementation of 1) role playing, 2) political battles, and 3) resolution using a GDSS in the IS curriculum is discussed.

ENHANCING THE POLITICAL AWARENESS OF INFORMATION SYSTEMS STUDENTS

Political skills are an essential talent needed by computer professionals of the 1990s. Many varied activities of the IS staff require such skills. For example, the requests for software application development often exceed the resources of the IS department. How does one prioritized these requests? One common solution is the steering team approach. But, what happens when the steering committee reaches an impasse?

Similarly, when an application development project crosses department boundaries, a committee is developed. And, while the systems analyst on the team may have specific viewpoints, he is often called upon to be the monitor of the committee. If the representatives from the departments involved have conflicting viewpoints about the project's development, how does the analyst maintain peace and keep the project on proper track?

In the systems development course under discussion, such a scenario is enacted. Role-playing is an exciting alternative to the technological detail associated with many IS courses. Through role-playing, a conflict is developed, and the students are directed to resolve the conflict. While the scenario changes from one semester to another, a typical example is presented below.

A state-supported university bookstore was recently investigated for alleged improper selection of suppliers. As the campus newspaper reported the activities of the investigation, it became the topic of the systems development course. How can the information systems department help eliminate the problem in the future? Students quickly suggested inclusion of a routine to prioritize the vendors. Further analysis of the prioritization routine provided the desired conflict in the classroom.

The procedure for prioritizing vendors for a manufacturing organization became the class project. The class was divided into four groups: 1) The Accounting Department, 2) The Manufacturing Department, 3) The Marketing Department, and 4) the IS Department. Each student was assigned to a department. Because all of the students were majoring in IS, the instructor felt that a little time was needed for them to adapt to their new roles. Thus, the students met in their four groups to determine what variables would be important to that departmental group in the prioritization of vendors.

Each group was charged with obtaining a list of variables important to their role in the selection of a vendor. Variable such as cost, delivery time, delivery reliability, and quality were suggested. However, the group also had to determine an objective method for measuring the variable. Once the instructor felt that each student had a grasp on his role, the four groups were disbanded.

One student from each area was then placed in a "design" group. Each team was charged with developing an objective means for selecting a vendor. This involved not only a determination of which variables should be included in the function but also the weight or importance of each variable. The students were told that they would be graded individually on how well they represented their functional area in the design group. The competition for grades quickly produced conflict and dysfunctional groups. A simulation of a political battle had been created.

A few groups indicated the lack of participation of one of the team members, so they eliminated his/her variables. In the effort to gain support for their departmental area (and for their grade), students were willing to jeopardize the quality of the final decision from a corporate standpoint. Selected conflict management techniques, such as liaisons, confrontations and negotiations, were used. These techniques did help reduce the intensity of some of he political battles, but the teams still had difficulty in reaching a unanimous decision.

Slightly less than one-half of the teams utilized a group decision support system package. This specific GDSS was primarily a mathematical model. Also, the students had access to one copy of a 'brainstorming' GDSS.

Based on their two page summary report, it was evident that the groups utilizing the GDSS voiced more satisfaction with their decision than those groups who elected not to use the group decision support system. Through the use of these software packages, and the strategies for conflict management, each group completed the project and wrote a satisfactory report. More important than the vendor selection routine was the written discussion of conflict resolution and group dynamics. The students appeared to have learned and internalized the concepts of group dynamics and conflict resolution.

CONCLUSIONS

Technology is growing at an alarming rate. Forecasts of the implications of such growth can be made utilizing past history and Nolan's stage model. Technology is changing the way organizations do business in the 1990s. It is affecting the balance of power and responsibility within the organization.

User departments are assuming some of the activities, such as user developed applications, of the IS department. However, user departments are finding it difficult to complete their original responsibility, plus incorporating the responsibility for new IS related functions. Organizations are looking to computer internships for at least a partial solution to some of their problems.

Technology is also a catalyst for many of the political battles within a organization. Managers and staff, fearing loss of jobs, seek to "empire build" and engage in "turf battles". Solutions to these problems are provided in the organizational behavior concepts of conflict management. These solutions include the issues of liaisons, negotiations, confrontations and additional training. However, the use of technology may also provide insight into the resolution of political problems. Group decision support systems can assist business teams in finding a solution to the "turf battles."

BIBLIOGRAPHY

1. Daft, R. L. (1986). *Organization Theory and Design* (2nd edition). West Publishing Company.

2. Lucas, H. C. (1989). *Managing Information Services*. Macmillan Publishing Company.

3. Nolan, R.L. (1979). "Managing the Crisis in Data Processing." *Harvard Business Review*. March/April,1979.

4. Nolan, R.L. and C. F. Gibson. (1974). "Managing the Four Stages of EDP Growth." *Harvard Business Review*. January/February.

5. Santosus, M. (1993). "The Hottest of the Hot Seats." Supplement to *CIO*. September 15, 1993.

6. "Software for Little Guys with Big Horizons." (1991). *Business Week*. March 18, 1991, p. 10.

7. Sprague, R. H., Jr. and B. C. McNurlin. (1993). *Information Systems Management In Practice* (2nd edition).

8. Stair, R.M. (1992). *Principles of Information System: A Managerial Approach*. boyd and fraser.

9. Zwass, V. (1992), *Management Information Systems*. Wm. C. Brown Publishers.

An Alternative Curriculum for Introductory Information System Students

Chris Cope and Pat Horan
Department of Computing and Information Science
La Trobe University, Bendigo
PO Box 199, Bendigo 3550, Australia
cope@redgum.ucnv.edu.au

Abstract

The difficulties with teaching introductory information system development to large groups of students at the undergraduate level can be related to the students' lack of awareness of the complex business environments in which information systems are developed. This paper describes a number of teaching techniques which can be used to provide an awareness of the business systems environment. Written portrayals based on observation of teaching sessions by students and external observers were used to evaluate the approach.

Avison (1989) describes the conventional lecture, tutorial and examination method of teaching and assessing information systems curricula. The difficulties associated with teaching introductory systems analysis and design to large groups of young undergraduate students using these conventional techniques are well documented in the literature. Aspects of the problem include:

Students' lack of awareness of the business environment

Information systems are developed within a business organisation rather than in isolation. A systems analyst requires not only knowledge and expertise in the techniques of systems analysis and design but also an awareness of the way organisations operate, how information is used in an organisation and how people interact and react to possible change. To adequately impart the technical skills of systems analysis and design to students, Mantelaers and Creusen (1990) feel the teacher and the students must share an understanding of how organisations operate. Students attempting undergraduate studies immediately after a secondary education generally lack this knowledge.

Limited resources preclude a practical approach to learning

Little and Margetson (1989) emphasise the uselessness of learning about something without being able to practise it. No amount of the verbal or written instruction typically provided with the conventional teaching approach can prepare an individual to apply the technical skills of systems analysis and design in the complex social and organisational setting of a business environment. Providing a learning environment where these skills can be effectively applied is beyond the capabilities of most tertiary institutions and their academic staff given the large numbers of students involved, the research pressures on academics and the physical resources available.

The previous education or inclination of the students

Fritz (1987 p.129) says '... computer science programs tend to attract technically-oriented, arts-shy students who are most comfortable learning specifics such as language syntax rather than generalities such as organisational behaviour or societal problems'. When these students are confronted with an information system development project they will encounter multi-faceted tasks such as interviewing staff, oral presentations to management and interacting with people who are fearful of change. The ability to deal with the integration of the technical and human aspects of a system's development cannot be taught effectively using the typical lecturer/tutor centred approach.

Poor feedback on the practicality of designs

The design of input screens, reports and the user interface for a computerised information system will involve students producing static designs using pen and paper or a computer screen. Osborne (1992 p. 259) points out a major weakness in this, 'Many of their good looking diagrams could not have been implemented and only by attempting an implementation would they have discovered this'. In a large group, introductory undergraduate program time and resources are generally insufficient to allow full implementation of designs.

The problems show that the combination of current teaching methods and students' poor knowledge and experience of the systems environment makes it difficult for students to develop and apply the required systems skills and understanding.

The ideal solution

The literature documents the use of long-term, real-world, group-based projects to overcome the difficulties of teaching systems analysis and design to undergraduates (for example Fritz 1987, Little & Margetson 1989, Schuldt 1991 and many others). This solution is not practical in the introductory information system curriculum. The reasons include:

- Undergraduate units commonly involve many students. Finding and supporting large numbers of projects is resource intensive and beyond the capabilities of most information system schools.
- The theory of systems analysis and design needs to be taught at an introductory level before any long-term project can be considered.
- Many of the people-interaction skills required of students in a group-based, real-life information system project are enhanced by maturity and life experience. At the introductory, undergraduate level students lack this development.

Long-term, practical information system development projects are clearly suited to the later stages of an undergraduate information system course. The problem of providing a practical approach at the introductory level remains.

The proposed approach

The approach used in the course consists of two components. Lectures provide theory and large group interactive role play exercises. Tutorials are used to provide practical application of the theory through the integration of learning techniques which require students to behave like a systems analyst with activities designed to provide the business organisation environment. A description of the learning techniques applied broadly across the tutorial component of the curriculum will be given, followed by detail on some of the specific activities.

All tutorials involve students working in groups with each activity including planning and de-briefing sessions which have to be reported on either verbally or in writing. The tutor facilitates the learning process by assuming the role of business manager responsible for the information system development. Each analysis and design team is questioned on the contents of its report until understanding at the non-technical level of a business manager is achieved. Team members are required to justify and defend their positions on any analysis and design decisions they make.

In conventional introductory systems analysis and design courses students learn various techniques in isolation from one another and the system. For example a conventional textbook exercise in form design supplies all required data and the student concentrates on defining the layout. The proposed integrated approach involves the student establishing the data requirements using a role play activity with other students before designing the form. This helps form an important link between development activities such as specification development and design activities in the students' minds as well as refining fact finding techniques. Such a link is often missing when students receive problems only in written form and assume no further analysis and definition of the problem are required.

Individual tutorial activities

Individual tutorial activities designed to achieve this integrated approach include:

Practical investigation of an existing information system

Experience shows that many young undergraduate students have no concept of the components of an information system or the processes performed by an information system. It is very difficult to teach about the development of an information system when the students have no concept of the end-product. A small information system which stores data on home contents for insurance purposes is provided to each group of students to investigate. The students are required to write a report identifying the components of the information system and the specifications it satisfies. This activity not only introduces students to the functions of an information system but demonstrates the link between initial specifications and the end-product.

Card database simulation
A simple conceptual understanding of a database is central to information system development. To introduce students to the concept of a database they are required to perform typical data maintenance and retrieval tasks on a poorly designed card file for a simple student record system. The tasks are necessarily repetitive because of the data duplication inherent in the design of the card. A discussion on this data duplication leads to the re-design of the card file, creating instead three different cards related by foreign keys as a means of overcoming the data redundancy. The production of indexes to enable access to the data in varying orders completes the development of the card database. Where initially the card database consisted of one card file, the students now have three card files and two indexes. During later tutorials students computerise the card database to further reinforce the database concept and to demonstrate the capabilities of the computer at managing repetitive, mundane tasks.

Continual case study
When introducing many, potentially confusing new concepts to students it is important to minimise the difficulties faced at any one time. This can be achieved by using the same case study with all concepts. After using the student record card simulation the students have an understanding of the system requirements. When new concepts are introduced the students can concentrate on applying the new skills (for example data flow diagrams) to a situation with which they are familiar.

Analyst role play
One of the first tasks faced by a systems analyst investigating an information system problem is to accurately and in detail describe the current situation. This description, which may be in written form or as data flow diagrams, acts as a basis for further analysis and design. Commonly the analyst will attempt to develop the description by tracing the flow of data through the system and identifying the processes applied to the data by using the most appropriate combination of fact gathering techniques. Experience shows that students have great difficulty coming to terms with the idea of a systems analyst attempting to describe the current situation as an initial stage of systems analysis. Given a practical interviewing situation they tend to immediately look for solutions to a problem about which they know very little.

To explore the role of the systems analyst early in the analysis phase and to provide practice at fact gathering, groups of four students role play an initial meeting between an analyst and the staff of a small fast food restaurant. Three students in each group are provided with details of their role and the manner in which they may be expected to interact with the analyst. The student playing the analyst receives details of an initial phone call with the client and the meeting arrangements. At the end of the role play the tutor instigates and directs a discussion among groups on the performance of the analyst. The activity is designed to give students experience of a fact gathering situation and by comparison of the approaches of various analysts highlight the role of the analyst and the line of questioning necessary to describe the current situation.

Real-life analysis and design assignment
The value of a real-life, long term project to the information system curriculum has been described above and the difficulties in implementing this learning technique with large numbers of students described.

An actor is co-opted to role play a client for an information system development project. A completed, real-life project is used to ensure as much authenticity as possible and to minimise possible contradictions. The same development project is presented to all students as assessable work. Working in teams, students initially meet the client in a lecture situation. The client explains the background to the project, overviews the problem and asks for questions. Individual students ask questions with the question and ensuing answer able to be recorded by all students. The teams of students have two further opportunities to question the client. This occurs in small-group tutorial sessions. Again all students in the group have the chance to benefit from each student's questions. Each team is required to produce the majority of a Systems Requirement Document for the client.

Documentation production

A comprehensive test of students' understanding of the components and processes of an information system is to require them to prepare the user documentation for a system still at the design stage. Familiarity with all concepts associated with the database, user interface, input screen and report design is necessary to produce a user manual in advance of system construction. Students must be able to visualise the information system in operation, a thorough test of the completeness of their designs.

A systems analyst needs to be able to communicate at various levels. Producing a user manual for a system, in addition to testing design expertise, requires explanation of an information system's functions at a relatively low reading level and in non-technical terminology. The ability to write clearly and simply is essential for a working analyst.

Students are required to prepare the user manual for the information system developed in response to the real-life situation used for assessable work and described above. Design is based on each team's systems requirements unless they are deemed to be unsatisfactory in which case a sample set of specifications is provided. The design and user manual production process is assessed rather than the intricate detail of the design.

Prototyping

This technique involves using the capabilities of a 4GL (fourth generation programming language) to rapidly implement, test on users and alter where necessary, designs for input screens, reports and menus. Students are able to experience interaction with users over design possibilities and also test the practicality of their designs, overcoming a major problem mentioned by Osborne (1992).

Walkthroughs

Behan (1991) uses structured walkthroughs in a classroom setting to assist students to learn to view their efforts from the perspective of others in order to avoid negative feedback. Students are required to review their analysis and design with groups of peers. The peers provide feedback on logic problems, practicality of designs and feasibility of the system.

Evaluation of the approach

Students appeared to react well to the curriculum. However a concern remained as the literature documents bias by teachers who design and implement their own curriculum (for example Davis 1980, Zeichner and Tabachnick 1982). Do the students really obtain the knowledge, skills, concepts and perceptions intended? An evaluation was necessary.

What evaluation model should be used in this situation? No precedent exists as the literature proposing new teaching methodologies for systems analysis and design curricula contains only gut-reaction evaluation. This evaluation was essentially for the teacher's benefit and so the method had to take into account the widely accepted demands and time constraints on teachers (Bayona, Carter and Punch 1990), the natural bias of self-evaluation caused through curriculum ownership (Davis 1980) and the insecurity teachers feel towards any form of evaluation (Wolf 1973). Borich (1989) believes that the majority of existing classroom observational tools are too extensive and cumbersome for use by classroom teachers. A curriculum evaluation method had to be designed specifically for this situation. The method consisted of an amalgamation of the self-reflective and portrayal approaches to curriculum evaluation (Scheyer and Stake 1976, Stake 1976, Kemmis 1977, Elliot 1978, Harlen 1978, Hall 1979, Kemmis and Hughes 1979, Davis 1980, Kemmis 1980, Smith and Fraser 1980).

Students from one tutorial group and external observers were used to provide the instructor with a non-judgmental, unbiased view of classroom activities. The external observers were chosen to provide complementary insights (e.g. educational perspective and information system development perspective). Observers wordprocessed their observations and e-mailed them to the teacher who prepared a written portrayal of each tutorial session based on the views of the observers. The contents of the teacher's portrayals were negotiated with one of the external observers until agreement on accuracy, relevance and fairness was reached. This technique was used by Fraser (1989) to validate the contents of qualitative portrayals as a means of eliminating teacher bias. The end-product of the evaluation is a document which represents a portrayal of the tutorial component of the introductory systems analysis and design subject. Interested parties, including the teacher can read this document and form their own opinion of the curriculum.

This method is designed to enforce structured self-reflection by the teacher on an unbiased view of the curriculum. The teacher is then able to pass judgment on their own curriculum based on the model proposed by Smith and Lovat (1990) to explain how teachers make curriculum judgments. Each teacher has a teacher-self frame which is made up of the teacher's unconscious beliefs about an ideal teacher. Beliefs about planning and implementing the ideal curriculum form the teacher's curriculum belief system. The teacher's beliefs about the ideal teacher are tempered by the perception of their actual teacher self and the closeness of this perception to their ideal teaching self. A disparity causes a limitation in the number of curriculum planning and implementation options open to the teacher. The interaction between the teacher's ideal curriculum and the curriculum they perceive themselves capable of producing creates the standard against which a curriculum evaluation will be judged.

The portrayal of the analyst role play activity is included as Appendix A to allow the reader to form their own judgment of the activity and to demonstrate the nature of the curriculum activities.

Evaluation conclusions

The evaluation method requires readers to form their own opinions on the tutorial activities. Several points stand out however:

- The portrayals confirmed the motivating effect of the tutorial activities on the students.
- Students readily adopted the group work and participated actively. They seemed to enjoy learning in this way.
- However, an overall strong group identity failed to materialise. This was probably due to the changing membership of the small groups formed for each activity and a response to encouragement from the tutor to constructively criticise each others' opinions.
- Despite the aim of each tutorial activity being presented at the beginning, both students and external observers failed to recognise the designed purpose of most activities
- Involvement in the evaluation process was well-maintained by both student and external observers and contributed to the students' development of interest and analytical attitude to the subject and their roles in it both as participants and observers.

Conclusion

A systems analysis and design curriculum was designed and implemented in an attempt to overcome a lack of relevance recognised in existing curricula at the introductory undergraduate level. The curriculum contained a business-oriented, practical aspect to overcome the undergraduate students' lack of knowledge of the real business environment. The tutorial component of the curriculum allowed students to practise combinations of skills in small groups in realistic systems development settings.

An evaluation of the approach showed that students were motivated by the approach and activities but often failed to relate the particular activity to the systems development context. This problem can be related to the stage of personal and professional development of the students. However the observed benefits of the proposed approach are sufficient to justify its further use and development. The skills and confidence gained should increase students' readiness for later information system units in their course, especially any long term projects involving a systems development life cycle.

Students educated in information system development with a practical approach from the beginning should be welcomed by employers. Any approach which gives students a better overview of the nature of information systems and improves the chances of new employees moving into team work and successful contact with users with minimal corporate training can only be of benefit to the information system industry.

Reference List

Avison, D.E., (1989). Action programmes for teaching and researching information systems. *The Australian Computer Journal*, **23**,2:66-72.

Bayona, E.L.M., Carter, D.S.G. & Punch, K.F. (1990). The Role of Teachers in Curriculum Development. *Curriculum Perspectives*, **10**,4,:9-19.

Behan, B.A., (1991). Developing thinking skills in systems analysis and design courses. *Interface*, **13**,3:37-39.
Borich, G.D. (1990). *Observation skills for effective teaching*. Columbus, Ohio: Merrill Publishing.
Davis, E. (1980). *Teachers as Curriculum Evaluators*, Sydney: George Allen & Unwin.
Elliot, J. (1978). Classroom Accountability and the Self-Monitoring Teacher, in Harlen, W. (ed.), *Education and the teacher's role*, Schools' Council Research Series, London: MacMillan.
Fraser, B.J. (1989). *Case Studies in Post-Compulsory Curriculum*. West Australian Social Science Education Consortium.
Fritz, J. (1987). A Pragmatic Approach to Systems Analysis and Design. *SIGCSE Bulletin*, **19**,1:127-131.
Hall, M.E. (1979). Portrayal as a way of addressing problems in communicating evaluation findings, in Fraser, B.J. (1986). *Annotated Bibliography Of Curriculum Evaluation Literature*. 2nd. ed. Perth: CUT.
Harlen, W. ed. (1978). *Evaluation and the Teacher's Role*. London: Macmillan.
Kemmis, S. (1977). Telling it like it is: The problem of making a portrayal of an educational program. In L. Rubin (Ed.), *Curriculum Handbook: Administration and theory*. Boston: Allyn and Bacon.
Kemmis, S. (1980). Program evaluation in distance education, in Fraser, B.J. (1986). *Annotated Bibliography Of Curriculum Evaluation Literature*. 2nd. ed. Perth: CUT.
Kemmis, S. and Hughes, C. (1979). Curriculum evaluation in higher education: Self-reflection in a critical comunity, in Fraser, B.J. (1986). *Annotated Bibliography Of Curriculum Evaluation Literature*. 2nd. ed. Perth: Curtin University of Technology.
Kemmis, S. and Stake, R. (1988). *Evaluating Curriculum*. Deakin University Press.
Little, E. and Margetson, D.B. (1989). A project-based approach to information systems design for undergraduates. *The Australian Computer Journal*, **21**,2:130-136.
Mantelaers, P.A.H.M. and Creusen, M.W.F.J. (1990). Teaching Information Systems Design: Mission Impossible? In McDougall, A. and Dowling, C. (Eds.) *Computers in Education*. Amsterdam: Elsevier Science Publishers.
Osborne, M., (1992). APPGEN: A tool for teaching systems analysis and design, *SIGCSE Bulletin*, **24**,1:259-263.
Scheyer, P. and Stake, R.E. (1976). A program's self-evaluation portfolio. *Studies in Educational Evaluation*, **2**:37-40.
Schuldt, B.A., (1991). 'Real World' versus 'Simulated' projects in database instruction, *Journal of Education for Business*, **67**,1:35-39.
Smith, D.L. and Fraser, B.J. (1980). Towards a confluence of quantitative and qualitative approaches to curriculum evaluation. *Journal of Curriculum Studies*, **12**:367-370.
Smith, D.L. & Lovat, T.J. (1990). *Curriculum: Action on Reflection*. Social Science Press, Wentworth Falls, NSW.
Stake, R.E. (1976). A theoretical statement of responsive evaluation. *Studies in Educational Evaluation*, **2**:19-22.
Wolf, R. (1973). How teachers feel toward evaluation. In House, E.R. (ed.) (1973). *School Evaluation: The politics and processes*. Berkeley, California: McCutchan.
Zeichner, K. and Tabachnick, B., (1982). The belief systems of university supervisors in an elementary student-teaching program. *Journal of Education for Teaching*, **8**:34-54.

Appendix A
C114 Tutorial and Assignment Evaluation
Early Tutorial - Week 4
Information Gathering Role Play

Background

The FastFood restaurant is owned and, along with a waitress, operated by Mr. and Mrs. Owner. Mrs. Owner complains of having no time with her children because of her bookkeeping and waitressing duties. Mr. Owner, the chef, employs a systems analyst to study the situation and suggest possible solutions.

One of the first tasks faced by a systems analyst investigating an information system problem is to accurately and in detail describe the current situation. This description, which may be in written form or as Data Flow diagrams, acts as a basis for future deliberations. Commonly the analyst will attempt to develop the description by tracing the flow of data through the problem scenario and identifying the processes applied to the data by using the most appropriate combination of fact gathering techniques. This tutorial involves groups of four students role playing the staff of the FastFood restaurant and the systems analyst at an initial group interview.

Experience shows that students have great difficulty coming to terms with the idea of a systems analyst attempting to describe the current situation as an initial stage of systems analysis. Given a practical situation they tend to immediately look for solutions to a problem about which they know very little. This tutorial is designed to give students experience of a fact gathering situation and by comparison of the approaches of the various analysts highlight the role of the analyst and the line of questioning necessary to describe the current situation.

Portrayal of Early Tutorial - Week 4

The tutor began with a brief overview of the role play before dividing the tutorial members into two groups of four and handing out a role description sheet to each student. The allocation of parts to individual students was on the basis of the tutor's knowledge of the students to date. In this all male tutorial group half the members ended up playing female roles. The pivotal systems analysts role was deliberately given to students who would tackle the task with enthusiasm but not necessarily expertise. One analyst was a repeat student and so had participated in a similar tutorial the year before.

Students were allowed ten minutes to consider and discuss their role with the same character in the other group. The role play description sheet gave detail of the character's activities in the running of the business, clues to their personality and how they could be expected to react to the analyst. The tutor spent time with the two analysts advising them to control the meeting and discussing how they might run the interview.

The two groups reformed and the analysts began their meetings. The role play was allowed to continue to completion without interruption. This took approximately twenty minutes. The students participated enthusiastically and responded to questions from the analysts appropriately although no adoption of role personalities took place. The approach of the two analysts was in contrast although neither initially outlined the purpose of the meeting or explained its place in the systems analysis process. The repeating student performed well in the analyst role generally pursuing the flow of data through the organisation although occasionally lapsing into discussion of the future. The other analyst spent considerable time establishing a long term relationship with the clients and then talked exclusively about possible solutions. Much inappropriate technical language was used by this analyst confusing the clients.

The tutor then initiated a general discussion. Each analyst outlined their approach and successfully described a component of the FastFood restaurant's operations. Other students were asked to describe and comment on the approach of the analysts. Having listened to each other's comments the analysts were asked if, given the benefit of hindsight they would have approached the interview in a different manner. The analyst intent on establishing a long term relationship conceded the information he had gathered was less technical and '*failed to seize the essence of how the business processed its accounts, customer orders and timesheets ...*'. The tutor concluded with an overview of the interviews and what should have happened. Importantly the analysts were thanked for their part in the learning process and were assured any criticism was not personal.

The purpose attributed to the tutorial by the observers was varied and can be best summarised by the comments of the observer experienced in information systems teaching:

The purpose of the tutorial could have been :

> *to give experience in interviewing*
> *to give experience in fact finding*
> *to give reality to a systems problem*
> *to bring out hidden complexity in an apparent straightforward task*
> *to show that interviewing and fact finding are complex tasks*

Most observers mentioned the worthwhile nature of the tutorial although the observer experienced in information systems teaching bemoaned the amount of issues raised by the tutorial and left unexplored, citing interviewing technique and fact finding skills as examples.

THE IMPACT OF A CURRICULUM CHANGE ON PERCEPTIONS OF MIS

Earl Chrysler
Accounting and Management Information Systems
California State University, Chico
Chico, California 95929-0011
916.898.4623 ECHRYSLER@OAVAX.CSUCHICO.EDU

ABSTRACT

This research compares and contrasts the behavioral expectations of business students before and after completing the traditional introductory MIS course and a redesigned course that stresses the characteristics of functional area business systems. The redesigned course showed significantly different changes in student perceptions.

BACKGROUND

In most business schools the introductory MIS course has undergone numerous redesign efforts. The reasons for the redesigns have varied - changing the programming language, changing the computing platform from mainframes via terminals to PC's, eliminating programming, etc. However, the basic purpose of the course appears to have remained that of a computer literacy course with a business orientation.

A few years ago the MIS faculty, with the approval of the College of Business faculty, redesigned the introductory MIS course with the objective of changing its orientation. Since it had been observed that many beginning Business students were skilled in the use of a word processor and/or electronic spreadsheet and/or a database management system, elementary concepts in these areas were removed from the course and reassigned to individual one-unit credit/no credit remedial courses. The introductory MIS course was then designed to stress business information systems. This meant that, aside from the introductory computer topics, the major concern was making the students aware of the logic of functional area computer-based systems. Included were such issues as the master files typically found related to each functional area's processing, data fields normally found in these files, the processing logic of various updating programs and strategic, tactical and operational level reports traditionally produced for the finance, marketing, production and other functional areas.

Previous research had found the following items of interest to MIS faculty regarding students of the introductory MIS course at California State University, Chico:

1. The students, as future users of computer-based systems, believed they should not be the dominant influence during the system design process;

2. They also believed the systems analyst should not be the principal force during the system design process;

3. Increasing the future users' knowledge about computers did not appear to affect their beliefs, but did appear to give them an additional feeling of confidence about participating in a system design project.

Of interest was the question of whether redesigning the course would have an impact on the students' perceptions. In order to determine what effect, if any, redesigning the course had on the students, the original research that led to the above findings was replicated using students who were being taught with the new course syllabus, topic outline and faculty member orientation.

PURPOSE OF THE STUDY

The purpose of the original study was to investigate the following questions:

1. What types of behaviors do future users of management information systems believe are appropriate for the user during the system design phase?

2. What types of behaviors do future users of management information systems believe are appropriate for the systems analyst during the system design phase?

3. To what extent do future users' perceptions of appropriate behavior for a user during the system design phase agree with line managers' perceptions of behavior appropriate for success in a line position?

4. To what extent do future users' perceptions of appropriate behavior for a systems analyst during the system design phase agree with staff managers' perceptions of behavior appropriate for success in a staff position?

5. Do future users feel confident about participating in the design of a computerized system?

6. To what extent are any of the above perceptions different before and after having an introductory MIS course?

The purpose of this study is to answer those same questions using responses from students who were taught in the course after it was redesigned, and to determine what effect, if any, changing the course appeared to have on the perceptions of the students.

METHODOLOGY

For the purpose of the original study, a questionnaire was designed which paired each of five inner-directed traits with each of five other-directed traits, and vice versa. A final pairing, of two other-directed traits, was included as a diversion but not considered in the scoring process. As a consequence, a respondent's total score consisted of M inner-directed preferences plus N other-directed preferences, where M + N was equal to twenty-five. The subjects were told the purpose of the study as it related to user perceptions and asked to select, in each pair of traits, the more appropriate behavior for a user during the system design phase. They were next asked to repeat the process, but now selecting the more appropriate behavior for the systems analyst.

After completing those two scoring tasks, the students was asked to respond to the following statement:

> "With my present knowledge, I would feel self-confident if it was necessary for me to work with a person from a data processing department during the design of a computerized system."

The student was to respond by selecting from the following list:

1. I very strongly agree
2. I strongly agree
3. I somewhat agree
4. I somewhat disagree
5. I strongly disagree
6. I very strongly disagree

Each student completed the questionnaire during the first week of the semester and then again during the last week of the semester. Only data for students for whom both questionnaires were present and complete were used.

The identical procedure was used in this replication study.

FINDINGS

In order to examine the questions at issue, tables were constructed and the responses of the groups of interest analyzed.

The initial study showed there was almost no change in students' perceptions after completing the course as to the behavior appropriate for a user during the system design phase, as shown in Figure 1. After completing the redesigned course, however, there was a marked change from having an other-directed orientation toward having an inner-directed orientation, as shown in Figure 1A. Since the role of the user during the system design phase is similar to that of a line manager, i.e., the systems analyst is the staff support person assisting the user who actually decides on the features the system should have, this development could be perceived as a positive change.

Figure 1
Perceptions of Behavior AppropriateFor User

	Before	After	Total
Very Strongly Inner-Directed	7 (6.5)	6 (6.5)	13
Strongly Inner-Directed	15 (18)	21 (18)	36
Inner-Directed	38 (35.5)	33 (35.5)	71
Other-Directed	40 (41)	42 (41)	82
Strongly Other-Directed	50 (49.5)	49 (49.5)	99
Very Strongly Other-Directed	38 (37.5)	37 (37.5)	75
Total	188	188	376

Chi-Square = 1.52 P = .91
( ) = Expected Frequencies

Figure 1A

Perceptions of Behavior Appropriate For User

	Before	After	Total
Very Strongly Inner-Directed	2 (3)	4 (3)	6
Strongly Inner-Directed	13 (16)	19 (16)	32
Inner-Directed	8 (12.5)	17 (12.5)	25
Other-Directed	32 (25.5)	19 (25.5)	51
Strongly Other-Directed	39 (36)	33 (36)	72
Very Strongly Other-Directed	17 (18)	19 (18)	36
Total	111	111	222

Chi-Square = 8.96 P = .11
( ) = Expected Frequencies

In the original study there was almost no change in the perceptions of students after completing the introductory MIS course regarding the appropriateness of analyst behavior, as shown in Figure 2. After completing the revised course, however, there was a visible shift and, once more in the direction that is most desirable. The students tended to believe the analyst should exhibit other-directed behaviors, as is appropriate for a person in a staff position, before completing the course. After completing the course the students moved to define even more strongly the need for the analyst to be other-directed, i.e., supportive and assisting the user in making decisions. This shift is evident in Figure 2A.

Figure 2

Perceptions of Behavior Appropriate For Analyst

	Before	After	Total
Very Strongly Inner-Directed	16 (13)	10 (13)	26
Strongly Inner-Directed	32 (30)	28 (30)	60
Inner-Directed	31 (33)	35 (33)	66
Other-Directed	35 (37.5)	40 (37.5)	75
Strongly Other-Directed	54 (54.5)	55 (54.5)	109
Very Strongly Other-Directed	20 (20)	20 (20)	40
Total	188	188	376

Chi-Square = 2.17 P = .83 ( ) = Expected Frequencies

Figure 2A

Perceptions of Behavior Appropriate For Analyst

	Before	After	Total
Very Strongly Inner-Directed	5 (4)	3 (4)	8
Strongly Inner-Directed	12 (11)	10 (11)	22
Inner-Directed	14 (13.5)	13 (13.5)	27
Other-Directed	28 (24.5)	21 (24.5)	49
Strongly Other-Directed	42 (341)	40 (41)	82
Very Strongly Other-Directed	10 (17)	24 (17)	34
Total	111	111	222

Chi-Square = 7.53 P = .18 ( ) = Expected Frequencies

In Figure 3 one finds the effect of the course in the original study on a student's feeling of confidence to participate in a system design project. One would hope to find movement from the "disagree" categories to the "agree" categories. While some is present, it is not strongly pronounced. In the case of the impact of the redesigned course, the shift is much more evident, as shown in Figure 3A.

Figure 3

Perceptions of Confidence to Participate in System Design

	Before	After	Total
Very Strongly Agree	9 (9.5)	10 (9.5)	19
Strongly Agree	33 (37)	41 (37)	74
Somewhat Agree	64 (71.5)	79 (71.5)	143
Somewhat Disagree	35 (30)	25 (30)	60
Strongly Disagree	35 (29.5)	24 (29.5)	59
Very Strongly Disagree	12 (10.5)	9 (10.5)	21
Total	188	188	376

Chi-Square = 6.64 P = .24
( ) = Expected Frequencies

Figure 3A

Perceptions of Confidence to Participate in System Design

	Before	After	Total
Very Strongly Agree	5 (5.5)	6 (5.5)	11
Strongly Agree	22 (22.5)	23 (22.5)	45
Somewhat Agree	42 (48.5)	55 (48.5)	97
Somewhat Disagree	17 (17)	17 (17)	34
Strongly Disagree	20 (13.5)	7 (13.5)	27
Very Strongly Disagree	5 (4)	3 (4)	8
Total	111	111	222

Chi-Square = 8.62 P = .12
( ) = Expected Frequencies

In the original study the movement of individual students was analyzed in an attempt to determine in more detail the changes in perceptions during the semester. In Figure 4 and 5 are shown the movements of individual student responses regarding the perception of the behavior appropriate for a user and an analyst, respectively. The reason the number of entries in these figures does not equal the number of students in the study is that if a student did not change his/her perception, no entry is shown in the table for that student.

The desired movement in Figure 4 would be from the other-directed categories to the inner-directed categories. This would imply that one would prefer to see larger numbers appearing in the lower left area, columns one , two and three and rows four, five and six. Another positive effect would be movement from more strongly other-directed categories to less strongly other-directed categories. The section of Figure 4 that reflects that type of movement is columns four and five and row six, and column four and row five. Neither of these areas appear to have the type of numbers that would support the belief that the desired change in perceptions had occurred.

Figure 4

Change in Perception of Behavior Appropriate for User

	AFTER					
BEFORE	VSID	SID	ID	OD	SOD	VSOD
VSID		3	2	0	0	1
SID	0		7	1	3	0
ID	0	5		12	7	5
OD	2	4	5		16	5
SOD	1	4	6	13		11
VSOD	6	1	5	5	10	

Figure 4A

Change in Perception of Behavior Appropriate for User

	AFTER					
BEFORE	VSID	SID	ID	OD	SOD	VSOD
VSID		0	1	0	0	0
SID	0		3	2	2	1
ID	0	4		2	2	0
OD	2	4	6		11	5
SOD	0	3	6	6		8
VSOD	1	2	1	3	5	

The desired movement regarding perceptions of behavior appropriate for an analyst would be large numbers in columns four, five and six, and rows one, two and three, which is movement from the inner-directed toward the other-directed categories. Also considered somewhat favorable would be movement from a strong inner-directed area to a lesser inner-directed area. This would be shown by large numbers, relatively speaking, in row one, columns two and three, and row two and column three. Once again, as shown in Figure 5, the data do not give evidence that desirable changes have occurred.

Figure 5

Change in Behavior Appropriate for Analyst

	AFTER					
BEFORE	VSID	SID	ID	OD	SOD	VSOD
VSID		2	6	3	1	2
SID	3		6	6	6	3
ID	3	8		6	9	1
OD	0	4	11		8	3
SOD	0	6	7	13		5
VSOD	1	1	2	3	6	

Figure 5A

Change in Behavior Appropriate for Analyst

	AFTER					
BEFORE	VSID	SID	ID	OD	SOD	VSOD
VSID		0	1	0	1	2
SID	0		0	4	2	2
ID	0	1		1	6	0
OD	1	1	1		13	4
SOD	1	4	4	7		10
VSOD	0	0	1	0	4	

The results of the redesigned course in terms of movement of individual students is shown in Figures 4A and 5A regarding the perceptions related to users and analysts, respectively.

The individual movements that are of primary interest regarding perceptions related to user behavior are found in columns one, two and three, and rows four, five and six. Of secondary interest are movements from a strong other-directed area toward less-strong other-directed categories. The area of Figure 4A of interest is rows five and six and columns four and five, and row five and column four. While these numbers are somewhat encouraging, a more dramatic shift would be desirable.

If the students had changed their opinions regarding the role of the user and the analyst, one would expect to see their changed perceptions regarding the behavior appropriate for an analyst as movement in the table. The movement one would desire is from inner-directed behavior to other-directed behavior. This would be shown in Figure 5A as relatively large numbers in rows one, two and three, and columns four, five and six. Additional support would be found in a strengthening of the feeling of appropriateness of other-directed behavior. This would be apparent by numbers in columns five and six and rows four and five.

As can be seen, there has been considerable movement from the inner-directed area to the other-directed area, which is desirable, in addition to some movement out of the weaker other-directed areas to the stronger other-directed areas.

When comparing Figure 4 to Figure 4A, it is readily apparent that there has been movement in both instances from the other-directed to the inner-directed areas. The computed chi-square values for Figure 1A versus Figure 1, which analyzes the data shown in detail in Figure 4A and Figure 4, however, imply the movement related to the redesigned course has been significantly more pronounced. Similarly, the computed chi-square values for Figure 2A versus Figure 2 support the position that the redesigned course has resulted in students changing their perceptions regarding the appropriate behavior for an analyst from inner-directed to other-directed.

SUMMARY AND CONCLUSIONS

During the design phase of a computerized system project, a user and an analyst jointly determine how a system will function. During this relationship the user must assure that the resulting system will meet the user's needs. It is the responsibility of the analyst to provide technical assistance by offering alternatives and advising the user on the capabilities and limitations of the hardware and software that will be used to generate the system outputs.

One could expect that a user would believe it appropriate to exhibit behaviors classified as inner-directed. The user could also be expected to believe that the analyst, being in a supportive role, should exhibit behaviors classified as other-directed.

After completing an introductory MIS course a student should be familiar with the roles of the players in the development of management information systems. The student should also have acquired sufficient knowledge of the MIS field to be willing to actively participate in such a design project.

This study compared the impact of teaching the introductory MIS course in two different manners. The first method was as a course taught with the dual objectives of being a computer literacy course and familiarizing students with business data processing fundamentals. The other method was with the introductory productivity software topics removed and redesigned to concentrate on characteristics of computer-based information systems related to the functional areas found in a typical corporation.

The study data supports the position that teaching the introductory MIS course with a concentration on the general area of functional area business information systems has a significantly different impact on student perceptions of the proper role of the user and the systems analyst during the design phase of system development and on their perceived self-confidence as to their ability to participate in a system development project.

The redesigned course results in students who report a significant increase in self-confidence regarding their qualifications for participating in a system design project.
Also, by stressing business system concepts it appears that during the course students change their impression from feeling that a user should be relatively other-directed to relatively inner-directed in interactions with an analyst. Further, during the semester the students also revise their beliefs regarding the appropriate behavior for an analyst from that of relatively inner-directed to relatively other-directed. Since the resulting beliefs are more appropriate for the roles of the participants in a system design project, it would appear that incorporating more business system topics in the introductory MIS course will more adequately prepare a Business student for his/her role in the corporate world.

DISTANCE LEARNING IN THE IS CURRICULUM

Kathryn O. McCubbin
Department of Management & Marketing
College of Business & Economics
Christopher Newport University
Newport News, Virginia 23606
804-594-7215 mccubbin@powhatan.cc.cnu.edu

Abstract

Many public and private institutions of higher education are under increasing financial pressure. In response to this pressure curriculum innovators are reaching to technology to develop cost effective teaching methods. On such method is that of distance learning: using telecommunications to deliver instruction. A small (3500 FTE) public four year institution has been experimenting with the use of the commercial *America On-Line* network service. A senior course in the Management Information Systems curriculum was selected since these students would be familiar with tele-commuting concepts. This paper describes the design of that course, the student reactions, and its pedagogical results.

BACKGROUND

The value of problem solving using text communication rather than traditional human communication has been documented in the research that has been done in the area of electronic meeting systems.[1] Studies cited by researchers at the University of Arizona indicate that electronic group meetings encourage a more equal participation among the participants and are a more efficient use of time than traditional meeting sessions.[2] In addition, removing subjective qualities of communication from group discussion removes much of the obfuscation that occurs regularly, by design or otherwise, from traditional human interaction.

Electronic meetings have been compared to teleconferencing which enjoyed popular appeal in the 1980's but whose utility appears to have waned. Teleconferencing allows people located in specially equipped rooms in remote locations to hear and see each other. Promoters of this technology emphasized the savings in travel costs. While those cost savings have not materialized, teleconferencing is viewed as an essential part of an organization's strategic plan. It uses the technology to do things it could not do before; it promotes frequent communication among group members

---

[1]Dennis, George, Jessup et al, "Information Technology to Support Electronic Meetings", *MIS Quarterly*, Volume 12, Number 4, December 1988.

[2]ibid.

thereby lessening the chance of error due to misunderstandings.[3]

The introduction of commercial on-line systems such as *Prodigy* and *America On-Line* (AOL) provide an electronic meeting facility to the general public at modest cost. This situation coupled with the fact that over 70% of the student body has access to a personal computer presented the faculty with a cost-effective opportunity to explore distance learning.

The course chosen for this experiment was Management of Information Systems, the capstone course in the Information Systems curriculum. Students are expected to be proficient in the use and design of business application systems. The goals of the course are to develop an understanding of how technology is used to manage information within organizational structures and to develop an understanding of how information and information technology is exploited for strategic gain. Mastery of this topic implies the ability to:

* describe the function of IS management at operational, managerial and strategic levels

* describe the leadership responsibilities of IS management

* describe the organizational impact of information systems

* describe emerging technologies and issues

The required texts were Information Systems: Management Principles in Action by Robert K. Wysocki and James Young and the accompanying casebook, A Collection of Management Situations. Optional texts included The Publication Manual of the American Psychological Association and The Official *America On-Line* (AOL) Tour Guide. AOL was chosen as the network provider because of its low monthly cost, its graphical user interface, and the offer of several free connect hours. The AOL software and ten hours on the AOL network were complimentary to all new customers. Access to a desktop computer which supports *America On-Line* software was required. Such support included an IBM 386/486 compatible computer, a mouse and a Hayes compatible modem.

COURSE DESIGN

The instructor requested AOL to send each student the *America On-Line* software before the semester began, informing the students by mail of this fact and asking them to install the AOL software as soon as they received it. Because of incompatible configurations two students had to share a personal computer. AOL provides three facilities which lend themselves to distance learning: private electronic meeting rooms for real time interactive sessions, e-mail

---

[3]Michael Hammer & James Champy, Reengineering the Corporation, Harper Collins, New York, New York, 1993, pp. 89-91.

and bulletin boards for asynchronous sessions. The class met for three hours each week alternating between traditional on-campus classrooms and electronic *America On-Line* meetings. E-mail for informal communication between students and instructor as well as among the students was encouraged.

During the first traditional class meeting on-line students exchanged screen names and were told to test their on-line access by sending e-mail to each other. The instructor advised them to use the address book feature of the software and told them the name of the private meeting room. The first on-line class occurred during the third class meeting. Part of the assignment for that meeting was a report on articles for the students' research papers. This report was to be e-mailed as an external file to all class members and the instructor 48 hours before the on-line class. Students were expected to download these files and be prepared to discuss the reports they contained during the on-line class.

Once the students were on-line in the private meeting room the half-duplex conversation began. The computer displayed the conversation of all parties in the room as comments and responses were received. Since responses were limited to 100 characters, the sender had to type <enter> and then resume typing if the message exceeded this value. While the sender was typing other students responded with unrelated comments. This caused the display of a fragmented version of the longer than 100 character response. Such confusion was remedied in subsequent sessions by using the following conventions: an exclamation point to indicate raising your hand, a question mark to indicate a question, three dots to indicate that you are continuing a comment on another screen line, and "ga" for the session leader to recognize a participant. Capital letters, sarcasm, and creative graphics were to be avoided.

It became quickly apparent that the assignment, the discussion of journal articles, was too open-ended for an on-line session. Much of the communication in a classroom is done with body english, facial expression, and other non-verbal signs. One of the ways to counter the absence of these is to structure the session. Accordingly, the subject of all future on-line classes were the case studies from the cited text. The case studies are rich in content and present controversial issues with a colloquial sense of humor. Students took turns in being the session leader, the originator of the "ga's". The format was well received and was maintained for the duration of the course.

The absence of non-verbal communication is not necessarily a disadvantage. The student is forced to depend solely on electronic text which causes them to focus their attention on what was being "said". It discourages mind-wandering. Furthermore, entire class session can be captured in a file by the instructor and downloaded for future study. Subjects that students had difficulty discussing could be identified and revisited at a later class. Individual student performance could also be reviewed at the instructor's leisure and filed. This was of significant value when an explanation of the grading process was requested by the students.

The bulletin board facility of *AOL* was used for asynchronous communication regarding the student research papers. Reports on two journal articles each week were posted to the bulletin board. The articles were to be obtained from searches conducted of electronic libraries on the INTERnet. Each student was asked to comment on at least one report from each of the other students. These comments were stored in a "thread" of the bulletin board and downloaded by the instructor for grading purposes. Although the use of the bulletin board facility was not as popular as the on-line experience, it did provide a hands-on example of how a variety of electronic resources could be integrated.

Many students enjoyed the network experience to the extent that they continued their subscription to *AOL* after the completion of the class. Others preferred to experiment with *Prodigy* and *Compuserv*. The use of distance learning was also a positive experience for the instructor. The major pedagogical advantage is the active participation of the students in the conduct of the class and the corresponding decrease in lecture format, although this requires that each class be prepared in detail before the beginning of the term. The use of distance learning is now being promoted aggressively by several departments.

Summary

Electronic meetings conserve the physical plant and financial assets of the institution. Fewer classrooms are needed. If two classes hold electronic meetings on alternate weeks, assuming they meet once a week, then one classroom can serve double the number of students. The campus network is not involved so no additional computing resources are required. The greatest advantage, however, is that distance learning using electronic meetings equips students with the knowledge and use of leading edge technology.[4]

Electronic meetings for use in the IS curriculum improve the course content and expose the student to leading edge technology. Parameters for successful implementation are: the number of students in the class, the sophistication of these students in on-line systems, student access to home personal computers, and the

---

[4]National Public Radio, "Talk of the Nation", WHRV Norfolk, Virginia, February 3, 1994.

support of the institution.

Domain Analysis is an Essential Skill of the OO Analyst

Les Waguespack, Ph.D.
Computer Information Systems Department
Bentley College
175 Forest Street, Waltham, MA 02154-4705
(617) 891-290 LWAGUESPACK@BENTLEY.edu

Abstract

Software reuse is a primary motivation in object oriented technology. *Class library* is perhaps the best representation of this idea. It can be a versatile and cost effective repository of reusable successful software development experience. It can save time and effort in succeeding software projects that fall into the same *application domain*. This potential for increased productivity and cost effectiveness holds a strong attraction for software development managers. The key is focusing class library development on an application domain rather than on a particular application. This paper discusses the potential and challenge of the application domain-based class library. It compares and contrasts its characteristics with those commonly found in software libraries based on earlier technology. Using class libraries as a form of software library can achieve effective software reuse. This requires a fundamental reorientation in development priorities and goals. And, domain analysis is at the root of this reorientation. This is a new educational challenge in the preparation of professional software developers.

0.0 Introduction

This paper argues the importance of domain analysis in the development and evolution of information systems based in the object oriented paradigm - therefore the importance of domain analysis in the education of information professionals. To promote this argument I first discuss concepts of reusability and survey the language features that have contributed to software library technology. Then the paper discusses software library as a tool of software development with focus on selected conceptual milestones. Object orientation is briefly defined with emphasis on the class declaration mechanism and how it is used to define a class library. This discussion is based on contemporary theory and practice within the object oriented paradigm[Wegner 90, Cox 84, Coad 92, Goldberg 83, Meyer 87, Rumbaugh et. al 91]. The paper continues by examining how the IS professional can exploit object orientation to achieve the software reuse goals. Object orientation may change the practice of software reuse. Finally, I will argue that the power of object orientation must be channeled through a disciplined application of priorities and objectives by designing a class library based upon *domain analysis*. Furthermore, the discipline indicated here does not have a natural counterpart in the practice of software library design of the past. It therefore requires a new and focused treatment in the education and training of IS professionals.

1.0 Reusability

Reusability is an intrinsic motivation in information systems. Indeed, the basic concept of a calculator which may be repeatedly reused with set after set of data is the computer's primary underlying paradigm. The quest for reusability is embodied in a myriad of ways in hardware and software technology [Krueger 92].

Reusability is directly related to levels of abstraction [Shaw 84]. Useful transformations of binary values are codified and encapsulated as programming language operators to be used and reused for a variety of similar bit level manipulations. Convenient models of values are defined as data types in programming languages to form an underlying basis of information representation which can be reused from problem to problem in program to program, and so forth. These examples not only illustrate abstractions, but also reflect the two tracks within which reusable abstractions have traditionally evolved in computing. These are procedural abstractions and data abstractions, respectively.

2.0 An Abbreviated Account of the Evolution of Software Library

A software library is a repository for program code representing data or program segments which a programmer expects to use again. A library may be as primitive as a file structure containing text or as complicated as a system including tools for depositing, managing, and retrieving code for inspection, editing, or input to a compilation system. Entries in the library may be considered stored abstractions with varying levels of sophistication and completeness. The nature of the entries in the library is left to the ingenuity of the programmer who deposits them there. The utility of the library may be judged by the variety of future uses[1] to which the library may be applied.

The earliest forms of stored abstractions were "built into" the programming languages themselves in the form of operators and data types. These abstractions formed the basic modeling components which the programmer assembled, sequenced, or programmed to construct software that satisfied the behavioral requirements of the application problem. Because these abstractions were intended to be intrinsic to a "language" there was a fair degree of conservatism on the part of the language designers with regard to the number of abstractions and the complexity of their functionality. Some of this conservatism was surely the result of the primitive nature of compiler design and the difficulty of constructing language translators which were among the most complex programs in the early days. Their inherent complexity limited the frequency with which a language could be enhanced by extending the abstractions it would support. It soon became desirable to support the addition of abstractions without modifying the basic language (and its translator). This was pursued through two general approaches: 1) the subroutine library, and 2) language extensibility.

2.1 Subroutine Library

It is possible to develop new programs by copying all or part of existing programs and altering the source statements to change the software's behavior to satisfy a new requirement. How easy this is depends largely on the programmer's familiarity with the base program, the similarity between the new requirement and the original requirement, and the difficulty with which reusable parts can be identified, isolated and extracted. In this mode a library of program source code may be considered a resource from which programmers may draw inspiration or actual code for future projects.

As another method, particularly complex sequences of programming (I/O control, complicated mathematical routines, etc.) can be written as subroutines with a defined entry point(s) and predefined input and output parameters. In contrast to reusing whole programs as above this approach concentrated on identifying potentially reusable code components in advance of a particular program's construction. These are placed in either source code or object code libraries for later access. If a routine is stored in source it can be copied and modified to suit whatever purpose. If a routine is stored in object form it can only be used "as is." To support an object subroutine library a programming language must provide a generic module invocation and linkage facility to reference those stored routines. By standardizing the module invocation and linkage facility it is possible for programs to utilize subroutines written in different source languages.

Although library routines may utilize other library routines, it was not a matter of course in early development systems. The abstractions represented were almost always purely functional in a form $Y = f(X)$. The design emphasis tended to focus on compartmentalizing function rather than on facilitating composition of function by combining the capabilities of multiple primitive routines.[2] As a result these libraries tend to be one dimensional. There are few interrelationships designed into the collection of library routines. The sequencing of processing and any hierarchical relationship among routines are the control responsibility of the invoking program which passes (sometimes large) chunks of data back and forth among routines.[3]

As reusable abstractions object code libraries provide two simple alternatives: use a routine as is or don't. Source code libraries offer the added option to copy a routine and modify it. This alternative is quite unconstrained and gives rise to a variety of

[1] (...future uses and <u>future users</u>...)

[2] Early object code libraries provided a mechanism for distributing enhancements to the programming language system through which they could be accessed. This allowed the languages themselves (and their translators) to enjoy a more moderate frequency of change. Most common subroutine libraries in use today serve the same general purpose.

[3] In fact, such interrelationships are often eschewed because the inter-dependencies among routines can present difficulties with some memory management schemes such as overlaying.

issues concerning consistency and reliability; in particular, the relationship between one generation of a routine and the next might be described as "accidental."

2.2 Language Extensibility

Language extensibility affects the ease and extent of abstraction that can be expressed in a programming language. Language extensibility is achieved by introducing facilities into the language that allow the programmer to extend the basic set of operators and/or data types by defining new abstractions that can later be invoked.[4]

2.2.1 Macros

Commonly in assembly language systems (and sometimes in higher level language systems) macros are used to generate source code by having the translator "replay" a source template and replace certain text in the template with parametric text provided when the macro is invoked. This technique allows the programmer to define a frequently used sequence of statements and have the sequence inserted into the program text at various points. By altering the source text of the program the basic design and function of the programming language and its translator are unchanged. In some cases, the macros are handled in a preprocessing step in the compilation job stream. By increasing the sophistication of the preprocessing step it is possible to accommodate more complex and powerful abstractions in the form of macros without affecting the complexity or stability of the underlying programming language.[5]

Macros are a common feature of assembly language programming systems. Macros have found new notoriety as procedural abstractions in spreadsheet applications.[6]

2.2.2 Data Types

All programming languages support some base set of data abstractions (some combination including integer, character, binary, floating point, record, set, etc.). Some languages allow the programmer to extend this base set of data abstractions by adding user defined data types. These types may then be used as templates when declaring variables. These facilities often allow the programmer to specify possible values in terms of upper and lower bounds, a subset of values from some other data type, or an enumeration of values with an explicit collating sequence. In its simplest form this allows the use of meaningful identifiers to represent variable states rather than modeling such a state with a representative base type value such as integer or character. In addition it provides a mechanical means for the language translator to detect inconsistencies where values of one type are combined or assigned to variables or values of a different, incompatible type. This effectively allows the translator to detect what otherwise may have been an undetectable semantic error as a simple syntactic one.

Another form of data abstraction provides data aggregation. The grouping of data elements into records or sets allows a collection to be named and manipulated as a whole. In most languages this is limited to transferring or copying the aggregation as a unit. Any functional manipulation tends to require that constituent data elements be manipulated individually as if they were not members of the structure.

Data types first appeared as language features in FORTRAN in the form of INTEGER, REAL, LOGICAL, etc. Structured types such as arrays, records, and files were staples of COBOL. The ability to define new data types became widely known through languages such as Pascal and Modula [Shaw 84].

2.2.3 Abstract Data Types

Abstract data types represent a fusion of features based upon procedural abstraction and data abstraction. A critical ingredient in most data abstractions is the set of operations that can be applied to data of a particular type. Simple data types define possible state values and the collection of data elements, but do not define the animating operations. Abstract data types allow the programmer to specify the content and structure of data along with the operations that may be used to change the state of that data. In its strictest form state changes on abstract data variables are only allowed through the use of its defined operations, a property which is referred to as *encapsulation*. Prior to abstract data types it was not possible to extend languages with modular constructs as complex as

[4] Although calling a library subroutine might be considered an extension of the operations available in the language it is qualitatively different from the extensibility described in this section.

[5] In practice macro languages may be as fully complex as the host language they augment by supporting extremely sophisticated and complex algorithms including selection and iteration.

[6] Macros in spreadsheet systems are almost universally interpretive rather than generative as in assembler systems, however they follow the same parametric structure.

stacks, files, or graphical interface components because such mechanisms were characterized as much by their available transformations as by their possible data states.

Abstract data types are exemplified in the language CLU. Ada's *package* construct is based on abstract data types, as is the *unit* construct which extended UCSD Pascal and is currently found in Borland's Turbo Pascal dialect. The concept of C.A.R. Hoare's *monitor* also encompasses the abstract data type definition [Wegner 90].

2.3 Software Library Summary

The subroutine library and language extensibility approaches are independently practical and complementary contributions to software technology. The subroutine library facilitates the transportation of programming from one problem to the next by acting as a repository facility to the programmers. Language extensibility allows the programmer to embed a degree of adaptability in a program segment that permits its reuse, but with some degree of modified behavior. By combining these two approaches software engineers have developed powerful tools to improve productivity and reduce development and maintenance costs.

The account presented here, "how the software library concept has evolved" is itself "abstracted." The concepts of procedural and data abstraction as well as the concepts of source and object routine library have been presented in a "homogenized" fashion. Concepts evolve in tandem with significant cross-interference and influence.[7] My intent is to lay out a framework of traditional use that software libraries have seen, a framework into which object orientation is being introduced.

3.0 Object Orientation

This section reviews object orientation as a system of concepts and language features. The characterization of object orientation presented represents a synthesis of descriptive material found in the literature [Wegner 90, Soustrup 86, Cox 84, Coad 91, 92, Goldberg 83, Meyer 87, Rumbaugh et al. 91].

[7] For example, the package facility in Ada includes a mechanism for parametric declaration of packages using a concept called *generic* packages. This effectively combines features of macros, typing, and abstract data types to facilitate the construction of libraries of reusable programming (facilities for I/O, data management, etc.).

Object orientation is based upon the concept of *objects* and *class* and the practice of defining these which exploits properties of encapsulation, inheritance, and polymorphism.

3.1 Object

An object is a uniquely distinguishable instance of an abstract data type which encompass a *state* (defined by *instance variables)* and a series of logical operations (called *methods)* that may manipulate that state. Objects are dynamic entities which are created, replicated, and destroyed.

3.2 Class

Each object is derived from a *class*. An object's *type* is defined by the class from which it is derived. Objects derived from the same class share identical structure (instance variables and methods), but do not share the same state (instance variable values). A class may be derived from another class. An instance of class is an object of that class as well as its ancestor classes.

3.3 Inheritance

A derived class *inherits* all the structure of its parent class and when it is defined may be extended with additional instance variables; it may add methods or substitute a new behavior for any or all methods it inherits. (A system which allows multiple parents for the same class is said to support *multiple inheritance*..) A class may not be a member of its own ancestry.

3.4 Messages

Interaction between objects is characterized by the passage of *messages*. A message names the receiving object, one of the receiver's methods, and possible data values to be sent in the message. The receiving object may or may not respond by returning data values. The particular way in which an object responds to a message is called a *behavior*. Collectively an object's methods define its overall behavior.

3.5 Polymorphism

An object of a particular class is also a member of any parent class(es) of that class. When a message that is sent to an object refers to an inherited method the inherited method's behavior is invoked unless the object's parent class has overridden that method in which case the overriding behavior is used. This property of behavior selection is called *polymorphism*. (In some instances method names may appear in classes without any common parent(s)[8]. In this case the receiving object would respond based on its defined behavior for that method name.[9] }

3.6 Encapsulation

In its purest form an object enjoys complete *encapsulation* which limits access or modification to its instance variables to only its own methods. Encapsulation hides an object's state information from agents beyond the object's own methods.[10]

3.7 Abstract Classes

It is possible to define a class for which there is no intention of instantiating objects. Such a class is designed for the sole purpose of facilitating the derivation of other classes. Classes of this type are called *abstract classes* and often include declared methods for which there is no defined behavior. It is the responsibility of the developer who derives a new class from this abstract class to define that behavior. Notice, however, that since the parent class defined the method, objects of that class may be sent messages referencing that method which will be routed properly according to the polymorphism that is in effect at that time. By such a practice several different behaviors for the same method name may be "naturally selected" by messages sent to a variable of an object's parent (or ancestor) class in which the method name is defined.

3.8 Class Library

A class library is a collection of classes for which every parent class or referenced class is a member of the collection. A class library is a system of abstractions whose interrelationships are designed to facilitate the composition of a range of program behavior that share the underlying functionality and abstractions of the library. A class library at a certain "high" level of abstraction describes a *domain.*

[8] They would always be related through the highest level parent, the root class OBJECT CLASS.

[9] A message to an object indicating a method that the receiver does not have results in a response to the effect that "I don't know how to do that!"

[10] Encapsulation is not unique to object orientation as demonstrated in the explanation of abstract data types. Furthermore, it may implemented in varying degrees such as allowing instance variables belonging to the class to be accessed by objects of that class, or by granting "shared" access by declaring the range of objects or classes that are allowed to directly access its instance variables.

3.9 Object Orientation Summary

The "flavor" of object orientation (that one develops a "taste" for) usually depends on the dialect of language or tool through which one articulates object oriented specifications. This paper is highly influenced by a consciousness of Smalltalk from which many object oriented concepts have flowed to other languages with object oriented features. In particular, all elements of software expressed in Smalltalk are defined in classes and implemented as objects.

4.0 Object Oriented Software Library

Object orientation impacts the software library concept affecting the subroutine library mechanism and the character of language extensibility. This section explores that impact and how it manifests itself in software library development and use.

4.1 Subroutine Library Mechanism Impact

The most obvious impact here is in translator complexity and sophistication. The incorporation of inheritance significantly complicates the task of the object oriented language compiler writer. Unlike simple text substitution in macros or simple external subroutine references and linkages to subroutine libraries, class and object declarations must be grafted right onto the syntax and semantics "tree" of the compiler's language definition.[11] It is common for the reference trace of an identifier to its declaration to require the traversal of 10's of levels of nesting of inheritance. This requires that the object protocols (the identifiers of objects and their methods accessible to clients) be compiled for each and every program compilation[12] which results in a significant increase in the number of defined identifiers that the program writer must circumnavigate.

In languages that support true polymorphism it is not possible to resolve all identifier references at compiler time. In particular, method references must be dynamically resolved at run time to support polymorphism. This run time activity contributes to some increase in program execution time.

Finally, the complexity of the interactions among objects both during compilation and at run time result in a complex testing and debugging

[11] Although some object oriented language technology has been initiated with preprocessor technology such as C++ to C translators, this does not seem to be an approach that will be long lived.

[12] Some systems also require that the entire class hierarchy be compiled in the session along with the client program.

environment which is virtually impossible to cope with without sophisticated interactive debugging support. The programmer who wishes to benefit from the library must become facile with an environment for searching, inspecting, retrieving, and testing classes and objects derived from class libraries.

4.2 Language Extensibility Impact

Because object orientation subsumes the character of abstract data types it is possible to utilize object oriented constructs to compose and use abstract data types without recourse to inheritance or polymorphism.[13] This would result with program segment declarations greatly similar to the subroutines classically found in libraries, primarily one dimensional, with very few (if any) interrelationships, and exhibiting very limited behavioral flexibility. The focus would remain on individual classes rather than on systems of classes and their potential interrelationships.

The power of object orientation goes beyond data abstraction which allows one to capture the state and behavior of individual abstractions. Object orientation allows the programmer to capture not only the state and behavior of individual objects, but the abstract composition of those objects as they are similar to and different from other objects for a collection of similar applications, an application domain. Just as a genetic engineer attempts to isolate and identify the various genes that distinguish an organism's makeup and behavior, a class library engineer attempts to perceive *patterns of behavior* in a situation within an environment along with the stimuli that evoke those behaviors. The result is a qualitatively different kind of library - a library that characterizes a collections of applications rather than one.

Where the entries in a subroutine library may be primarily independently complete in their definition and construction, class library entries tend to be more fragmentary and component oriented in their form. Where reading and understanding the documentation of a single routine in a subroutine library will probably allow its correct use, class library entries are more likely to be useful in concert with other class entries. To understand the function and use of one class will probably require a study of its function within a related group of classes. In this way a class library represents a fundamentally more complex concept than a subroutine library. This complexity comes into effect not only during the programmer's use of the class library, but also in the class library engineer's endeavor to craft it.[14]

4.3 OO's Promise for System Modeling

Object orientation promises to be a significant step (leap) in the evolution of expression of abstraction in software systems. That promise lies in the possibility of capturing abstraction that is a more fundamental explanation of the function of software systems in relationship to the real world. This is completely consistent with the evolution of programming languages that we have witnessed in the past. At each point where we achieved the technology to translate more abstract real world characterizations into specifications that can be operationalized through computer programs, we have evolved our tools to model that real world. Object orientation represents another evolutionary step in programming language technology, the stage is set for another step in modeling paradigms.

I offer here a very small first step in the modeling paradigm evolution. It is based on the form of contemplation that seems appropriate when attempting to a craft class library. It seems centered on the mind set that is taken when attempting to understand the application domain to be modeled[15]. I characterize it by offering three definitions of analysis orientation. The first two are commonly in practice today. The third seems to be in practice with those designing class libraries and object oriented systems. First let's consider two Webster definitions:

> **behave**: 1: to bear or comport (oneself) in a particular way, 2: to conduct oneself properly.

> **behavior**: 1: the manner of conducting oneself, 2 a: anything that an organism

[13] This could be likened to avoiding and user defined types and only using base types in Pascal or Modula.

[14] Perhaps the best example of this complexity lies in the decision to place behaviors in a class hierarchy. Polymorphism depends on the availability of defined methods at a level higher than the type of the object being referenced. This requires that the class engineer predict the breadth of interest that such a behavior will have and define (at least the interface of) that behavior high enough in the ancestry to allow messages to a "family" – messages that polymorphism will sort out by invoking the appropriate behavior for the specific object involved. Once defined, migration of the behavior upward has a ripple effect on the protocols of all the descendants. Furthermore, the more distant the level of the defined behavior from the object, the more abstract that behavior tends to be; the more abstract, the less obviously relevant the behavior to the "real world."

[15] This is in distinct from the modeling of a single application which is often the focus of programming language based treatments of object identification and specification as in [Coad 92b, Nerson 92, Rubin and Goldberg 92].

does involving action and response to stimulation, b: the response of an individual, group, or species to its environment, 3: the way in which something (as a machine *[and as software]*) behaves.

Now consider these three ways of looking at the application domain:

Procedure driven : that disposition in thinking that focuses initially and primarily upon the actions observed in a situation within the environment and then proceeds to determine the data which must flow through and among these actions to completely model the situation.

Data driven : that disposition in thinking that focuses initially and primarily upon the data components of a situation within the environment and then proceeds to define all actions in that environment that set and change those data components.

Behavior driven : that disposition in thinking that attempts to perceive patterns of behavior in a situation within an environment along with the stimuli that evoke those behaviors and then proceeds to discover the state data and actions that must be present to sustain those patterns of behavior.

I believe that it is true that early subroutine libraries both facilitated and promulgated *procedure driven* analysis and design. The advent of language extensibility and more advanced compilation technology permitted and encouraged *data driven* analysis and design. The stage now seems set with the growing maturity of object oriented programming languages that *behavior driven* analysis and design are possible.

4.4 Domain Analysis

The evolutionary change represented by behavior driven analysis is as much philosophical as technological [Fichman 92, Korson 92, Shlaer 88, Ward 89, Wegner 92]. In essence the orientation toward perceiving patterns of behavior focuses more so on the *domain* in which the problem being analyzed resides than on the problem itself. The behavior, interrelationships of data and control flow through messages, reflects patterns that are intrinsic to the domain of the problem and therefore are at a higher abstract level than the problem itself.

The notion of abstract class, for instance, is a characterization of a conceptual behavior that transcends the instances of objects found in a particular problem and provides a platform upon which to define a family of problem solutions within the same domain. The most common real world example of this "platform concept" is the Graphical User Interface framework provided as an environment for interfacing application programs into a particular execution environment. Windows™ or the Macintosh Toolbox™ class libraries define a domain of graphical behavior which may be specialized to the particular needs of an application.[16]

A domain analyst attempts to discern the patterns of behavior that transcend the specific problem at hand. In so doing the analyst defines a framework that is the foundation of reuse in object oriented systems development. This is a qualitatively different form of analysis than has been classically included in the education and training of IS professionals. It is more closely akin to the process of programming language design. It requires a greater facility with the manipulation of abstract concepts and specifications and a deeper knowledge of the environment "surrounding" the application being developed.

4.5 OO's Challenge to IS Analysts and Management

Behavior driven thinking is dramatically different from procedure driven or data driven thinking because it assumes that the resulting specification will not only reflect a specific application's model, but also the application's orientation in a broader application domain model. That is where reusability may be exploited. If class libraries are to exceed the capabilities of software libraries of abstract data types they must capture the application domain model.

I conclude that this difference in focus on the part of the software library designer is critical to the potential of class libraries over software libraries. Such a difference will not be served by simply adopting an object oriented programming language; it

[16] The specialization is facilitated only within the range of behavior defined in the framework. In this sense the framework also represents a discipline which normalizes the behavior of applications within the framework environment.

must be preceded by a philosophical reorientation in the analysis and design approach which is employed.

Classically trained IS professionals are not prepared to conduct this type of analysis and design. They have been carefully conditioned to focus on applications rather than upon application domains. They are most likely to see software resources as collections of programs rather than an aggregate model of an organization's information processing. IS professionals will have to be reoriented in order to achieve the promised benefits of reusability through object orientation.

This reorientation must include revised education and training of information systems analysts and designers, the incorporation of priorities for reuse in IS plans and procedures, and the establishment of IS functions specifically targeted to the development, security and enhancement of domain class libraries over an above the management of application class libraries.

5.0 Concluding Remarks

Application domain modeling is a mostly unresearched problem area. Clearly we can look at specialized programming languages targeted for particular application domains and say that we have some experience attempting to define useful languages for limited domains.[17] This represents a very limited experience.

The questions that arise are "Can we expect the level of analysis that leads to a language design to be expended on any application domain?" and "Is it feasible and affordable?" I believe the answer is "Yes!" when that commitment is integrated into organizational priorities and plans.

In the case of graphical user interface systems the answer is clearly, yes! All of the leading GUI's (Windows, Visions, MacApp, X-Windows)[18] are based heavily on object oriented technology and are delivered via class libraries. Research must begin to demonstrate that we can model and deliver other application domain libraries (relational database, compiler design, communications, hypertext, financial accounting, manufacturing, and so on).

Object oriented concepts represent more than an extension of abstract data types. They require a philosophical reorientation in the practice of software library design and use. Such a philosophical change can only come about by reorienting IS analysis and design education and training, and interjecting management priorities for reusability into IS plans and procedures.

[17] 4GL's are a good and successful example of such domain focused language efforts.

[18] References to the class library documentation for these various consumer available products.

References

[Coad 91] Coad, P. and Yourdon, E., *Object-Oriented Analysis* , 2nd Edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[Coad 92] Coad, P. and Yourdon, E., *Object-Oriented Design*, Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

[Coad 92b] Coad, Peter, 'Object-Oriented Patterns," *Communications of the ACM*, Vol. 25, No. 9, September 1992, pp. 152-159.

[Cox 84] Cox, B.J., "Message/object programming: an evolutionary change in programming technology (in IEEE Tutorial: Object-oriented computing, vol. 1: concepts, p. 150),"*IEEE Software*, Jan 1984, pp. 50-61.

[Fichman 92] Fichman, R. G. and Kemerer, C. F., "Object-Oriented and Conventional Analysis and Design Methodologies," *IEEE Computer*, Vol. 25, No.10, October 1992, pp. 22-39.

[Goldberg 83] Goldberg, A. and Robson, D., *Smalltalk 80: The Language and its Implementation*, Addison-Wesley, Reading, Massachusetts, 1983.

[Korson 92] Korson, T. D. and Vaishnavi, V. K., "Analysis and Modeling in Software Development," *Communications of the ACM*, Vol. 35, No. 9, September, 1992, pp. 32-34.

[Krueger 92] Krueger, Charles, W., "Software Reuse," *ACM Computing Surveys*, Vol. 24, No. 2, June 1992, pp. 131-183.

[Meyer 87] Meyer, Bertrand, "Reusability: the case for object-oriented design,"*IEEE Software*, Vol. 4, No. 2, March 1987, pp. 33-42.

[Nerson 92] Nerson, Jean-Mark, "Applying Object-Oriented Analysis and Design," *Communications of the ACM*, Vol. 35, No. 9, September 1992, pp. 63-74.

[Rubin & Goldberg 92] Rubin, Kenneth S. and Goldberg, Adele, "Object Behavior Analysis," *Communications of the ACM*, Vol. 25, No. 9, September 1992, pp. 48-62.

[Rumbaugh et. al 91] Rumbaugh, James, Blaha, Michael, Premerlani, William, Eddy, Frederick, and Lorensen, William, Object-Oriented Modeling and Design, Prentice-Hall, Englewood Cliffs, 1991.

[Shaw 84] Shaw, Mary A, "Abstraction techniques in modern programming languages," (in IEEE Tutorial: Object-oriented computing, Vol. 2: Implementations, p. 146)

[Shlaer 88] Shlaer, S. and Mellor, S. J., *Object-Oriented Analysis: Modeling the World in Data*, Yourdon Press, Englewood Cliffs, New Jersey, 1988.

[Stroustrup 86] Stroustrup, Bjarne, *The C++ Programming Language*, Addison-Wesley, Reading, Massachusetts, 1986.

[Ward 89] Ward, P. T., "How to Integrate Object Orientation with Structured Analysis and Design," *IEEE Software*, March 1989, pp. 74-82.

[Wegner 90] Wegner, Peter, "Concepts and paradigms of object oriented programming,"*OOPS Messenger*, Vol. 1, No. 1, August 1990, pp. 7-87.

[Wegner 92] Wegner, Peter, "Dimensions of Object-Oriented Modeling," *IEEE Computer*, Vol. 25, No. 10, October 1992, pp. 12-20.

Use of Post-Implementation Analysis in Undergraduate Principles of MIS Classes

Nancy J. Johnson
Management Information Systems
Metropolitan State University
Minneapolis, Minnesota 55403
612.373.2743 nancyj@msus1.msus.edu

ABSTRACT

Designing student assignments that integrate theoretical constructs with real world applications is challenging, yet critical for reaching mature learners. In Principles of MIS, students have been conducting post-implementation audits on installed information systems and rate it highly as a learning experience. It integrates interviewing, networking, analytical, writing and speaking skill sets, and provides some unexpected benefits.

INTRODUCTION

The Principles of Management Information Systems class is designed to introduce students to the theory and practice of developing and managing computer based information systems in organizations. The challenge of conducting the class lies in enhancing the prescribed methods presented in the text book with lecture, case discussions and real world experiences in order to provide the students with a complete view of the MIS theory and practice.

The course has a pre-requisite of an introductory course in computer terminology and concepts. This course is required for MIS and Accounting majors and recommended for CIS and business majors. The class is designed for students who are juniors. In addition to the lectures and readings, the course includes an assignment for each student to conduct a post-implementation audit of an information system that was implemented in the past five years. Each student selects a project based on accessibility to participants for interviewing. Each student writes an analysis and makes a short presentation to the class. The assignment has been very highly praised by students for its real world application of principles learned in class and for the personal experience to learn more about an organization.

I have taught the class since 1982 at the University of Minnesota and since 1991 at Metro State. The majority of students who have taken the course are adult learners who are already employed and in the process of completing a degree. While there are several instructors of this class, a common format of assignments and texts are used in all sections. For this paper, I surveyed three sections of the class taught at Metro State in spring of 1994 (total of 64 respondents) with a questionnaire given during the last class meeting. I have also cited comments and experiences with the project from students in prior classes.

SURVEY RESULTS

In the survey, the respondents' average age was 33 years, and the gender was evenly balanced between males and females. 83% of the respondents were not MIS or CIS majors. This is expected because of many more students in the accounting major than in the MIS/CIS major.

Most students select projects in the organization where they work (71% did so in the survey). Other students select projects in non-profit organizations (a church, a day care center, a volunteer group, a recreation organization such as trap shooting, or the university) or in an organization where they have a contact (family or friendship) or formerly worked. The students work individually, but have worked in teams of 2 or 3 if the project was exceptionally large.

For most students, this is the first time they have done a post-implementation survey (only 18% of the respondents had conducted post implementation audits before.) 55% stated that this project taught them a great deal, 41% said it helped somewhat and only 4% stated that it didn't help them at all. Students cited learning more about their organizations, networking within the organization, and seeing the application of MIS theory most frequently as being the greatest benefit of the project.

Students come to the class with widely varying levels of writing ability. For many, this paper is the first time they have to write in a business/consultant reporting style. A short, two page paper is assigned after the first class meeting to provide the instructor with a device to evaluate the student's writing and editing abilities. Immediate feedback on the paper convinces the student of the instructor's mandate for good writing and editing techniques. The instructor can also use this tool to refer the student to writing tutors if needed.

The students turn in the four paper sections during the quarter at assigned times-providing several benefits. First, it prevents the student from writing the entire paper to the last week of the term. Second, it allows the instructor to give constructive feedback early in the process on content, direction, scope and style/editing. Third, the sections are 3-4 pages which makes it easier for the student to write them (a 10-15 page paper is very intimidating for many students). While this process increases the reading/editing time of the instructor, the excellent quality of the final papers more than warrants the time investment.

Students interview at least four different stakeholders in the project (a designer, a user, a manager and another person), in order to get a balanced assessment of the project's success. Students frequently report that this process enhances their networking in and understanding of the organization. It also highlights the wide disparity of opinions of success or failure. Many students state that interviewing strangers is a personal growth experience and enhances their confidence in their own abilities. Students frequently say they learned much more about their own job in the context of the organizational processing of the information.

In response to the survey question on what they learned from the assignment, 32% of the respondents stated that they learned how important political dimensions are for project success, 33% learned how critical communications/training are for success, 23% cited inclusion of all system users as being important and 13% reported learning how critical planning is to project success.

Each student makes a short presentation on the major findings of his/her analysis. For all too many students, this is the first time they have had to make a stand-up presentation. Each student is expected to have professional and appropriate overheads and to stay within the time limit. The style of presentation is that of a consultant making a report to the organization's executives. The presentation not only gives students a chance to practice speaking, the reports give the rest of the class real world experiences to see the many different styles of project management and implementation utilized in organizations (successful and not).

Students frequently report that they learned more about their own role in the organization and how the rest of the place works. They expand their networks within the organization and often see other opportunities for themselves in other jobs. Some students report that they are viewed differently by others in the organization after completing and presenting their report (more valued due to increased abilities and knowledge). One student said that her manager was so impressed with her analysis, she was sent to the Amsterdam office to make the same presentation.

In the survey, 92% cited the project as helping them professionally and 96% as helping them personally. Some respondents noted that they had not tackled this large a project before, nor had they talked to individuals in the organization that were "higher up" than themselves. Most frequently, gaining a broader perspective of the organization was noted. Many saw their own roles as more important than they had realized. 90% felt the project provided a better learning experience than a library research paper would have. Integration of theory with application was the strongest reason for this, and is consistent with general adult learner needs.

Of the 6% of the respondents who didn't like the project or did not feel that they benefited from it, they cited disliking the time the interviews took and having to go into an organization in which they didn't work. This is a very insignificant number of students, more than offset by the positive evaluations from the rest of the class.

Growth in self esteem, improved interviewing and writing skills, practice in making presentations and gaining a better understanding of their own organizations (and their role within it) were the most commonly cited benefits of the project. For some students, the project pushed them to a higher level of involvement and performance than previously achieved.

Overwhelmingly, the respondents said that they learned more about human nature, politics in organizations and the intricacies of systems design and development than they could have from lectures and the textbook. It is a significant realization for them to see that the text can only present "ought to do" scenarios and procedures, and that any prescription has to be judiciously applied to less than perfect human organizations.

One student reported that the process of analyzing the system that she had designed in her former organization helped her to realize that the closing of her branch office by the organization's out of state headquarters didn't reflect negatively on the quality of her own work. Another student used her paper in a portfolio of work she submitted in her application for a controllers job (which she got).

One student rewrote her current job responsibilities after she convinced the company owners that they needed a formal systems manager. She learned about the job functions in class, and used her analysis paper to convince the owners that they needed to address on-going systems management issues to correct the deficiencies in design and implementation.

In one project, the student team pointed out that all knowledge about the system resided in a single programmer, and there was no back up or documentation, making the organization very vulnerable. An unexpected outcome was that the programmer took the report and negotiated a significant raise with management before agreeing to train a back up programmer in the organization.

44% of the survey respondents reported sharing their analysis with the organization. Several reported specific actions from their report - hardware upgrades, increased training, and changes in management of other projects to incorporate lessons learned from the analysis. One person reported that the system training program was completely redesigned after realizing how much confusion existed.

Of those respondents choosing not to share their paper with the organizations analyzed, reasons cited included defensiveness of interviewees, fear of openly criticizing projects and lack of interest by their managers. These reasons parallel the reasons why very few post-implementation audits are ever conducted as part of the systems development life cycle.

SUMMARY

After using both library research assignments and this project for over ten years in teaching this class, I have found this project to be the most valuable for students' personal and intellectual growth. It provides a vehicle for the students to evaluate a real information system design and implementation, and to learn from the actions of others. Benefits such as identifying new job opportunities and expanding personal networks also make the post-implementation analysis a valuable learning experience.

Excerpt from **PRINCIPLES OF MIS** course syllabus:

Term Paper: The term paper will be a presentation of a real world case study of a computer based information system design and implementation. The paper will be written in the style of a post-implementation audit report. The content of the paper will come from student interviews with project participants, and will incorporate the writer's analysis of how the project could have been done better. The purpose of this paper is for the student to learn from the experiences of others.

Picking a project: The type of systems project that is appropriate for this paper is one that was completed within the past five years. It should be some type of information system with multiple stakeholders. The installation can be a successful one or a disaster. The student will need to interview at least four different stakeholders to learn their respective viewpoints of how the project was designed, built and implemented. Stakeholders can be system designers, programmers, user managers, data entry staff, people who use the information from the system, senior management, etc.

The paper will have several sections which the students will turn in during the quarter. The purpose of dividing the paper into sections is so that the instructor can give on-going feedback in order for the student to produce a final paper that is of the highest possible quality. Only the final paper will be graded, not the first three sections. If confidentiality is an issue, the names can be changed to protect the innocent, after discussion with the instructor.

Section 1	Proposal: A two page proposal of the case to be presented and analyzed. This will become the introduction section of the paper. Describe the organization, the problem the system was designed to fix, and the approach taken in managing the project.
Section 2	Case description: This three to four page section will go into more depth about the situation, the players, the tactics, the strategies, etc. This will be a very factual reporting of the project from the viewpoints of the individuals who were interviewed.
Section 3	Case analysis: This three to four page section will be the student's analysis of how well the project went, whether it met the original goals, and how it could have been done better. This is a highly subjective section and should incorporate principles learned during the course and in readings.
Section 4	Summary: One page summation of the major findings of the case.
Final Paper	All 4 sections, revised, and smoothly integrated into a coherent paper.

NOTE: Students should retain their original interview notes in case the instructor requests them.

Student Presentation: Each student will make a 7-10 minute presentation on the major findings of their term paper research project. Good speaking principles are expected - good organization, appropriate overheads or handouts, effective use of hands and eye contact, and staying within the time limit. In the presentation, the student should describe the organization and the project in the introduction, then describe the main events and milestones of the project, and the major points of the analysis of the project experience. The speaker should concentrate on lessons that can be learned from the experience.

THE ELEMENTS OF MULTIMEDIA COMPUTING

Dr. Sorel Reisman
California State University, Fullerton
Department of Management Science/Information Systems

After almost 20 years, Western society is finally beginning to feel the effect of personal computing. Although it had long been forecast, few prognosticators had the insight to portend that the phenomenon of *multimedia* would be the catalyst that would truly popularize personal computers. Today, multimedia computing (MMC) is on the minds and lips of almost anyone who uses computers, from chief information officers to elementary school children. In many ways this is really fascinating since the term "multimedia" was largely unknown only two or three years ago. It would come as a surprise to many evangelists of MMC that multimedia computing has a history that reaches back almost 30 years, almost to when the industry of computing was just beginning.

Despite that history, information systems (IS) professionals are largely uniformed about the "elements" of multimedia. In fact, many professionals are just beginning to realize that multimedia is very pervasive among home computer users, far more so than among businesses. In many instances, IS professionals are becoming aware that they require information about multimedia, information that includes its history, applications, technologies, and most important, its likely impact on their own IS activities. In concert with their needs, IS departments in business schools are also seeking ways in which to introduce MMC into their curricula.

This tutorial will provide an overview of multimedia computing, beginning with a review of its history, through to a brief description of some of MMC's enabling technologies. The second part of the tutorial will demonstrate some of the media that comprise the media of multimedia. MCPS (Multimedia Computing Presentation System), a Windows-based system that can be used to teach about multimedia, will be used for the demonstrations.

MCPS is an easy-to-learn and easy-to-use multimedia application development/presentation system for students or users with very little previous computer experience. Despite its simplicity, MCPS has been used to develop professional applications for kiosks, training, and information presentation using the media of color, text, graphics, audio, and full motion video. Since students do not require a high degree of computer literacy to use MCPS, the system can be used at any level in the curriculum, either in a standalone multimedia course, or as part of an IS course that focuses upon elements of information systems.

UNDERGRADUATE PLACEMENT TRENDS OVER THE LAST FIVE YEARS AND THEIR EFFECT ON CURRICULUM ISSUES

PANELISTS

C. Joseph Sass
University of Toledo

Dr. Denise McGinnis
Mesa State College

Dr. Inge Klopping
Bowling Green State University

Dr. Barbara Dennison
Wright State University

The continuing and rapid pace of change in information technology and its effect on the job market continues to force IS programs to review their curricula frequently. Because of these on-going transitions, it is difficult for universities to keep current with the numerous changes that are occurring. The business community expects colleges of business administration to lead the way in information technology. With the ever increasing number of possibilities for major courses in the curriculum, educational institutions must make difficult decisions about what topics to include in their IS programs.

To aid in the modification and development of new curricula, each of the panelists surveyed graduates of their respective schools in order to obtain job placement information and possible trends in the marketplace in an attempt to determine the proper mix of course offerings.

This panel will report the results of a number of surveys, which reflect the outcomes from different geographic regions and different time frames, beginning in 1988 and continuing on to the present. The panel will also present a proposed curriculum that provides for both a PC emphasis track and a mainframe emphasis track, with a minimum number of courses.

Fostering Collaborative Learning in Computing Courses

Carol Chrisman
Luann Stemler
Apllied Computer Science Department
Illinois State University
Normal, Illinois 61761
(309)438-5543 CACHRISM@ILSTU.EDU
(309)438-3228 LSTEML@ILSTU.EDU

Abstract

This article suggests several different types of activities for collaborative learning that can be used in computing courses at either the secondary or post-secondary level. Five different types of collaborative activities that could be used in the classroom are: group programming assignments, reviews or walkthroughs, group presentations, group writing, and brainstorming or evaluation tasks. This paper also discusses both the benefits and potential problems from having the students work in teams in these types of activities.

Introduction

In today's team-oriented work environment, the ability to work effectively in groups is an essential skill for job success. Each employee must be a team player actively contributing to the team's achievements. Working effectively in groups requires good communication skills. To prepare students for this type of future, they need experience with group activities so that they become familiar with how groups operate and the roles to expect in a team.

Although collaborative activities are common in some subject areas, learning activities in computing courses are predominantly individual assignments. Within university computing courses group projects are often included in systems oriented courses such as an Analysis and Design or a Database course [2,10]. Yet the feedback received from potential employers consistently suggests that students need more exposure to group activities.

Many teachers want to provide appropriate opportunities for collaborative learning but lack experience in designing group activities. Teachers are also concerned with the large investment of time and effort that will be needed to incorporate group activities and may not perceive the benefits as outweighing the costs.

A number of factors have made some types of collaborative activities possible that would have been much harder, taken significantly more time, and resulted in less satisfying results in the past. For example, it has become much easier to put together a good looking presentation. Faster and more powerful computers, networks, and increasingly sophisticated software tools can support a variety of work activities that are carried out in groups.

This paper will suggest collaborative activities appropriate to a variety of computing courses, including those taken by beginning computing students. Regardless of whether a computing course is oriented toward programming or productivity tools, collaborative activities can be utilized.

Types of Activities

Five different types of collaborative activities will be described in this paper. These activities are: group programming assignments, reviews or walkthroughs, group presentations, group writing, and brainstorming or evaluation tasks.

Group Programming Assignments

Most programming done in the industrial world is done in teams where it is common for a team to develop a large system, often taking a year or more. While it is probably not feasible or desirable in an introductory programming course to have one large group project be the only programming activity in the course, it is feasible to have some collaborative programming assignments.

There are several ways of organizing group programming assignments. For example, an assignment could have each team member responsible for one sub-module of a program. Two person teams could divide into roles of coding and debugging where one person specializing in writing the source code and the other testing the finished code. Any assignment where the number of programming pieces to be developed is less than the number of team members will promote a collaborative work style. A team could develop an integrated system with each member responsible for individual programs which had to work together. Advanced computing courses frequently have semester long team development projects [3,12].

Stemler and Chrisman [9] describe the use of two group programming assignments over the course of a second language programming course. The assignments consisted of developing a single program that was large enough to benefit from modularization. Individual teams were responsible for developing their own specific program design and determining the parameters for each module. Each member of a six-person team was responsible for building at least one module and testing its content. After writing the individual modules, they were put together and the team debugged the program. The team only had access to one copy of the team's entire program. On the first assignment, none of the programs ran when the modules were put together; they had problems with parameters not matching. The teams had failed to agree on the order and format of the parameters for each module and so individuals had made independent decisions and coded their modules accordingly.

Stemler and Chrisman [9] reported that on the second group assignment, which required more complex logic, the overall design and hierarchy charts improved. From their earlier experience, the student teams had recognized the need for planning. However, they didn't discuss the design/logic needed for each module and this program required some complex mathematical logic to be applied at several stages of the program, so they had logic bugs. Students found it very difficult to debug someone else's logic.

Reviews or Walkthroughs

Another type of team activity that is frequently used in industry is a review or walkthrough of some systems project deliverable. For example, teams often conduct code walkthroughs where the source code for a specific program is reviewed for errors (in logic or language), adherence to company programming standards, and to verify that the specifications are met. This type of review has been found to significantly improve the quality of the resulting system and to reduce debugging time (at the cost of the time for the review). Almost every analysis, design, or implementation deliverable can be the subject of a review. To improve the accuracy and completeness of the system, it is common to conduct walkthroughs of a system's data model and intended database design, of the system's planned functionality and system/program design, and of the interface design.

These types of reviews can also be conducted within academic computing classes. Student teams can be required to conduct code reviews before attempting to test an individual module or integrate a set of modules. When development teams are small, two teams can review each other's work. One of the authors has had experience with two person teams trading reviews of data models in a database course. The quality of the data models submitted by each team improved dynamically and the students have commented they think the activity is worth the time invested.

Having to explain their thinking to the review group will give a student valuable experience and hopefully improve that student's communication skills. Since the instructor would not usually be present at this type of review, students should feel more comfortable pointing out problems. Reading someone else's code and having to understand someone else's logic can be a valuable learning experience. Students often comment after such an activity that it was really surprising to see how different someone else's design and thinking was.

Group Presentations

One type of collaborative learning activity that can be used in any computing course is to have group presentations on topics relevant to the course. For example, a micro-computer course could have presentations that recommended a hardware and software solution to satisfy a small business's information needs. Another example is presentations in an introductory course on typical uses of computers in a particular type of industry (such as health care, manufacturing, or education). The organization, research, writing, and creating of visual aids that go into making a presentation offer a number of small tasks that can give a team a good opportunity to be successful in building a team atmosphere. Since students have different talents, this type of activity can let the group capitalize on each member's strength. Due to the increased availability of presentation and multi-media software (eg. Linkway, Storyboard, Powerpoint, Freelance Graphics), students can also get experience in using such packages for high quality and impressive presentations.

Group Writing

Another type of collaborative learning experience that is similar to group presentations is a group writing activity. Student groups create a report and use word processing or desktop publishing to produce a professional looking result. Specialized software for groups is even available to allow work on the same document. However, the main benefit would be the collaborative activity of determining the content of the report and planning, researching, writing, and reviewing each of its sections.

Brainstorming or Evaluation Tasks

A frequent activity that groups must perform is to brainstorm or evaluate possible solutions to some problem or task. Group Support software is now available to provide computer support for these types of group activities. But even without computer support, a valuable collaborative activity can be compiling a list of possible solutions to some problem and determining the advantages and disadvantages of each solution. This type of activity would frequently be combined with a group report or presentation, but would still be valuable with just a minimal list or ranking of the results as expected outcome rather than a full report.

Putting Groups Together

A number of authors [1,3,5] discuss issues critical to successfully managing student teams. Team selection is one of the most important issues the instructor must consider. It is often hard for students to work together, both because of personalities and because of time conflicts, and it's hard to encourage cooperation when students are competing for grades. Many students have had little or no previous opportunities to work on a group activity.

Teams can be chosen by three general methods: students choose their own group, random assignment to a group, or by the instructor selecting the students for each group. When students choose their own group one problem which can arise is that the higher potential students will cluster together leaving the less capable students to compose a single group. If teams are chosen early in the semester, before students get to know one another, the team composition usually mirrors the seating configuration or previous acquaintances of the students in the room. Random assignment to groups should insure a more equal distribution of talent to each team but does not address the personality conflicts which could arise.

According to Bast and Teeples [1], when the instructor forms the teams there are two basic considerations: interpersonal dynamics and the work-related skills of prospective members. Interpersonal dynamics means looking at the individual members to see if they will get along with each other. The second area, considering work-related skills, means consideration of the students' skills and whether team members' strengths compliment each other. The instructor can gather some of this information through observation, interviews, or by using a questionnaire. A third factor which must be considered is whether the members of a team have similar time schedules, so they can schedule team meetings.

Bast and Teeples [1] provide a guide suitable for distribution to students about working in groups. Their goal is to enable students to optimize group performance and individual satisfaction. They introduce students to techniques which help them organize their groups.

Grading

Deciding how to grade group activities is always a challenge. A frequent approach is to base an individual team member's grade on the instructor's evaluation of the team end products, peer evaluations completed by each team member, and the instructor's evaluation of the particular team member [2]. Unless their grade really depends on the quality of the team's product, students may not be motivated to make their team work. It is important to have the students evaluate their team experience. Spruell and LeBlanc [8] found there is an advantage to making students' assessment not only a feedback mechanism but also a learning activity.

It is highly beneficial to get student feedback on their perceptions of group activities. This can be done though a feedback report which can be anonymous or a graded assignment. Student feedback may even provide some helpful suggestions for the next time the instructor will do a similar or the same group activity.

Potential Problems

There are several potential problems that can occur with collaborative learning activities in computing courses. Pournaghshband [5] identifies nine problems encountered in team projects: 1) poor communication among members, 2) poor leadership 3) failure to compromise, 4) procrastination problem, 5) integration testing problem, 6) lack of cooperation, 7) lack of confidence, 8) conflicts in students' schedules 9) members' personal problems.

Lack of communication between group members is frequently reported as a major obstacle. In most cases, communication within a group is adequate only in the sense that all members of the team know what the others are working on. The goals of the team and detailed individual abilities are not usually spelled out. Team members can be unwilling to listen to suggestions of different methods of designing a solution. This can cause major problems within a team. It is very hard to overcome years of singular thinking strategies and to develop a group identity and way of thinking.

Lack of leadership is another common problem. After the fact, student team members often realize that their teams would have benefitted from choosing a better leader. A team leader can promote listening to all team members equally, organize the workloads, and help team members to inter-relate, all important tasks if the teams are to operate smoothly.

Instructors will face inevitable personality conflicts and unequal work contributions by group members. Bast and Teeples [1] suggest that one of the biggest problems a group can face is what to do with a problem member. Bast handles this problem by having every group develop a process to deal with a "Free Rider". Some suggestions are to put the member on a written notice; make attendance a requirement at group meetings; decide on a procedure when to reinstate the "Free Rider"; decide on a procedure when the "Free Rider" will be fired from the team.

Benefits

There are a number of benefits that computing students can gain from collaborative rather than individual learning experiences. Moncada [4] reports that cooperative learning is an alternative teaching strategy that fosters the development of interpersonal communication effectiveness skills. Wilson [11] reports that collaborative work can enhance problem solving performance for novice programmers.

Most students feel group activities are worth the effort. Researchers agree that cooperative learning can produce positive effects on achievement [6]. Students recognize this type of assignment prepares them for the "real world". Students can recognize the need for better communication. Spruell and LeBlanc [8] claim that developing interpersonal competence along with the mastery of subject content is particularly relevant for the computer student.

Students realize a sense of team responsibility in group assignments. Each student has the feeling of ownership toward their contribution. They quickly realize that when their own component fails, they were letting down the entire team. At the end of a group activity, one student commented that the old cliche "United we stand, divided we fall" also applies to team activities.

In group programming assignments, students also benefit from the exposure to other people's code and to the debugging techniques of others. It is common in industry for team members to help each other debug programs but this is not always encouraged in academic settings. Students can learn valuable lessons about how the coding style affects the readability of programs from this type of activity. Students often learn best from examples, both good and bad, and these assignments provide a rich source of examples.

There are a number of benefits in terms of the students' exposure to interpersonal group dynamics. Smart [7] found group experiences introduce students to real-world frustrations and the conflicts inherent in working relationships within groups. Group interaction in these types of assignments is an integral part of the learning process. Students will indeed experience frustrations typical of real group situations from these collaborative experiences.

Summary

Collaborative learning activities can be utilized in every computing course to help prepare students to be valuable team players. Working in groups can be a positive experience but can also cause many problems. This paper has suggested some types of group activities that can provide appropriate collaborative learning opportunities. With a small investment of time, teachers can foster collaborative learning that will have many benefits.

References

1. Bast, Karin and Joseph Teeples, "How to 'Work Smart - Not Hard' In MIS Project Groups", Proceedings International Academy for Information Management, December 1992, Dallas, TX p. 149-150.

2. Chrisman, Carol and Barbara Beccue, "Evaluating Students in Systems Development Group Projects", SIGCSE Bulletin 19:1 (February 1987), pp. 366-373.

3. Kurtz, Barry and Tom Puckett, "Implement a Single Classwide Project in Software Engineering using Ada Tasking for Synchronization and Communication", SIGCSE Bulletin 19:1 (February 1990) pp. 6-11.

4. Moncada, Susan, "Cooperative Learning in the IS Curriculum", ISECON '92 Proceedings of the Ninth Information Systems Education Conference, October 1992, Nashville, TN, p. 16.

5. Pournaghshband, Hassan, "The Students' Problems in Courses with Team Projects", SIGCSE Bulletin, 22:1 (February 1990), pp. 44-47.

6. Reglin, Gary, "The Effects of Individualized and Cooperative Computer Assisted Instruction on Mathematics Achievement and Mathematics Anxiety for Prospective Teachers," Journal of Research on Computing in Education, 22:4 (Summer 1990), pp.404-412.

7. Smart, Regina, "Introduction to Group Dynamics: A Critical Factor in System Analysis and Design Courses", ISECON '92 Proceedings of the Ninth Information Systems Education Conference, October 1992, Nashville, TN, p. 40.

8. Spruell, James and Louis LeBlanc, "A Course Planning Method to Incorporate Collaborative Learning in Information Systems Courses", Journal of Information Systems Education 4:2 (Fall 1992), pp. 6-11.

9. Stemler, Luann and Carol Chrisman, "Introducing Teamwork Through Programming Courses", ISECON '93 Proceedings of the nth Information Systems Education Conference, November 1993, Phoenix, AZ.

10. Tellep, Andrew, "Information Concerning Group Processes is Beneficial to Students in a Systems Development Course", ISECON '90 Proceedings of the Eighth Information Systems Education Conference, October 1990, Chicago, IL, pp. 7-11

11. Wilson, Judith, Nathan Hoskin, and John Nosek, "The Benefits of Collaboration for Student Programmers", SIGCSE Bulletin 25:1 (March 1993), pp. 160-164.

12. Wolfe, James, "Getting the Most out of Large Programming Projects", NECC'89 Proceeding, (June, 1989), pp. 314-317.

The Applications Development Cluster of IS'90:
Introducing Database Concepts

Dr. Roy J. Daigle
Mrs. Janet J. Kemp
School of Computer and Information Sciences
University of South Alabama

Abstract

The teaching of applications development has evolved into a software engineering based approach. The DPMA IS'90 model curriculum recognizes a critical need to introduce data modeling concepts in the Applications Development Cluster. In this paper we describe how this might be accomplished.

Introduction.
The University of South Alabama is a state-supported institution serving the Gulf Coast region surrounding Mobile, Alabama. There are approximately 8000 undergraduate and 4000 graduate students enrolled in nine academic units including Arts and Sciences, Engineering, Business and Management, and Medicine. The School of Computer and Information Sciences (CIS) is one of nine academic units at the university. CIS majors choose one of three areas of specialization: Computer Science, Information Science, and Computer Engineering. A specialization in Information Science consists of course work geared to the design and implementation of computer-based information systems. Classes are taught on a quarter system. Because of the urban nature of the University, it is not unusual for completion of a course sequence to be interrupted for various reasons: participation in cooperative education or internship programs, employment demands, or personal responsibilities.

In 1992 the Information Science faculty initiated a redesign of the IS curriculum [Daniels et al, 1992] based on the IS'90 model curriculum [Longenecker, Feinstein, 1991]. A major component of this redesign involved a three quarter course sequence for Applications Development. The issue of integrating Project Management concepts in the Applications Development cluster has been addressed in a previous paper [Daigle, Kemp, 1993]. The focus of this paper is integrating database concepts into this cluster.

<u>DPMA IS'90.</u> The IS'90 curriculum [Longenecker, Feinstein, 1991] is preseented as a collection of seven knowledge clusters within four areas of knowledge content. Each cluster consists of related concepts and activities that define the expected educational experiences for the cluster. Applications Development is one of the knowledge clusters, occurring early in the curriculum after completion of the Concepts cluster.

The IS'90 curriculum is based on a body of knowledge. The knowledge units are mapped to the knowledge clusters. Using a descriptive language similar to that of Bloom [Bloom, 1956], IS'90 [Longenecker, Feinstein, 1991] presented a mapping that also specifies a *target learning objective* for the interaction of a knowledge element with clusters.

<u>Database Concepts in IS'90.</u> Table 1 shows the concepts, activities and the prescribed exit learning level for Database Concepts in Knowledge Cluster D - Application Development from IS'90. In IS'90 [Longenecker, Feinstein, 1991] these concepts are extended into other clusters. IS'93 [Longenecker et al, 1992] and IS'94 [Longenecker et al, 1993] continue this recommendation. Specifically, in Cluster E - Systems Development, the student will use the database concepts to analyze, design, and implement complex information systems. Moreover, in Knowledge Cluster F - Systems Project, the student will be expected to apply this knowledge to more realistic problems, those that are beyond the typical classroom setting.

Our Approach.

Previous Implementation. In the past, database concepts and terminology were chiefly the exclusive domain of the Database Management course in most curricula. Although the infrastructure appears naturally in the Applications Development Cluster, formal presentation of databases concepts and terminology was delayed until the third course in the sequence.

Present Implementation. Our plan for introducing Database Concepts into the Applications Development Cluster begins in the first course of the three course sequence.; it builds upon and makes use of project management concepts as described in [Daigle and Kemp, ISECON '93].

In the remaining sections of this paper, we will

- describe the basis for integrating database concepts into the Applications Development Cluster,
- describe the application of the spiral learning principle of IS'90 [Longenecker and Feinstein, 1991]); proceeding from the introduction of database concepts in the first course of the cluster, through their reinforcement in the second course, and the impact on the third course (database management) and, lastly,
- list conclusions and observations from the initial effort.

Implementation Plan. Using survey results, Covey [1991] provides evidence that people want meaning and purpose in their lives - they wish to be managed by principles. This same conclusion applies to students for the acquisition of knowledge in their courses. Our approach to integrating database concepts and terminology is based on the following principles.

1. *Student assignments provide a means of introducing new knowledge elements while reinforcing existing ones.* The model curricula IS'90, IS'93, IS'94 identify Where and at What learning level a knowledge element is incorporated into the curriculum. No direction is provided as to How to accomplish this incorporation. Student assignments are designed to contribute to application level learning [Bloom, 1950] for specific knowledge elements. In the present approach, we make use of student assignments to integrate database concepts and terminology into the Applications Development cluster.
2. *A group can be more effective in their work when there is a shared vision and a direction established.* Covey, [Covey, 1991], uses the metaphor of a compass to emphasize that without a shared vision and an established direction, it is not possible for all members of an organization to aim in the same direction. In the present approach, the shared vision for the Applications Development cluster consists of a schema and a menu for a system of applications involving that schema. The instructors are responsible for providing guidance for the direction as defined by the curriculum model. Once the shared vision and direction are established, they are referenced and revisited through the three course sequence.
3. *A group has more commitment to a shared vision when the group participates in developing that vision.* One way to encourage this commitment is through a process called *Participatory Development* [Gronbaek 1993]. [Daigle, Doran, Feinstein, Vest 1993] define Participatory Development as a methodology for providing continuous opportunity, encouragement and motivation for all involved to directly impact the interim decisions and therefore, the final product of the process. In the present approach, students participate through class interaction in the development of the shared vision. The instructor guides the class through the process, acting as a facilitator of learning. Decisions made by the class during the process determine the semantic meaning of the schema.

Database Concepts in the First Course.

The Applications Development Cluster consists of a three quarter course sequence in the Information Science curriculum. Although the first two courses presently use COBOL both on an IBM mainframe and in a personal computer environment, the primary focus is on analysis and design of applications involving data and file structures. This focus permits migration to any suitable tool for implementation. Students entering the sequence have completed the

Fundamental Concepts Cluster modeled according to the ACM's Computer I and II [ACM, 1991]. Elementary skills for use of a DBMS are also acquired in a prerequisite end-user course in personal computer tools (IS'90, Cluster A; IS'93, Cluster B).
The integration of database concepts in the first course of the cluster consists of

- performing data analysis,
- characterizing and representing data models with E-R diagrams,
- using database terminology , and
- identifying relational operators.

The process used to achieve this integration involves three phases:

- first, developing a shared vision, then
- prior to implementation, generating a benchmark for validation of the project using a DBMS, and finally
- during implementation, characterizing the project in terms of relational operations.

From the first course where the database concepts and terminology are introduced through the third course where they are implemented with a relational DBMS, the terms and ideas are repeatedly referenced and described.

Developing a Shared Vision. The means of accomplishing the integration of database concepts and establishing a shared vision begins with the identification of a problem class. Figure 1 shows a representation of one possible problem class as a VIEW of four entities: A, B, C, and D. The VIEW is presented as a single entity class for which data maintenance and reporting are required. The process of analyzing the entity class for maintenance and reporting possibilities provides sufficient complexity to define terms and to introduce database concepts. Once the final data model is derived, a menu skeleton for a system of applications is developed. The menu skeleton provides opportunity to address integrity issues.

The advantages of the schema of Figure 1 include simplicity of design and availability of all data in one entity class. Guided by the instructor, the students uncover major disadvantages which include maintenance and reporting requirements. With continuous student participation and instructor guidance, an entity-relation object based model is derived. The student participation results in an evolution of Figure 1 to the schema of Figure 2. This evolution provides the opportunity to introduce and explain entity class, entity, E-R diagrams, and terminology such as relationship, entity integrity, foreign key, and referential integrity. The schema of Figure 2 is part of the shared vision. This E-R diagram symbolizes data association for projects in all three courses.

Once it has been developed, the schema is analyzed for entity class maintenance and the resulting impact upon relationships. With the schema as a guide, the possibilities for type and means of information extraction are also examined. The functional and reporting requirements for a system of applications are documented for future reference. This is shown in Figure 3 and becomes part of the shared vision. The menu skeleton represents what is to be done with the data model of Figure 2.

This phase occurs at the start of the course and requires about 5-6 class hours. Instruction and commitment are accomplished because students are involved in the analysis and development of the schema and the menu skeleton. Once the schema and menu skeleton are distributed, each individual possesses a copy of the shared vision. The overall direction of the applications development sequence has been established - implementation of the system specified by the menu skeleton for the schema. In the first course, movement towards the shared vision is achieved by implementing a single report on the report submenu for entity class D. In the second course, movement is achieved by implementing the entire menu system using COBOL. In the third course, movement towards the shared vision is achieved by implementing the entire system using a DBMS.

Developing a Benchmark for Validation. In addition to the data analysis process, the first course also uses a relational DBMS to develop procedures to validate output [Daigle, Kemp, 1993]. The students are required to implement the schema of Figure 2 by creating and providing data for the entity classes. In terms of the schema and the menu skeleton, the project requirements of the first course are established as a control break application involving entity classes B, C, and D. The DBMS assignment is to use the report writer capabilities of the DBMS to prepare a report form for entity class D only, establishing the control break results. These results then become the output standard with which students compare the output from their project. Thus the DBMS is used to emphasize the importance of testing and validation of projects and programs. This DBMS assignment will be revisited in the third course.

Characterizing the Project in terms of Relational Operations. Database terminology and concepts are revisited periodically during the actual implementation of the course project. Because of time constraints, only sequential file organization is presented in this first course. The project is a control break application which is characterized as the JOIN of entity classes B, C, and D over the keys of B and C and their respective foreign keys embedded in D. The JOIN is achieved piecewise in memory by processing D as a sequential file and using Table Load/Search procedures of entity classes B and C. This method provides a standard for comparison to an indexed approach taken in the next course.

Participatory Development in the First Course. Active student participation in the development of the shared vision is an especially critical component of the first course. In the first course of the sequence, the instructor guides the students through their discovery of the schema and the menu system. The schema evolves from discussion, it is not merely presented to the students.

Database Concepts in the Second Course. Consistent with the spiral learning principle described in IS'90, database concepts are revisited in the second course in the cluster. The integration of database concepts in the second course consists of the following:

- Revisiting the shared vision,
- Comparing methods for 3GL implementation,
- Introducing rudiments of the relational, hierarchical, and network database models, and
- Rethinking the original problem in terms of VIEWS and embedded SQL.

Revisiting the Shared Vision. The schema and menu skeleton are derived during the first class meeting. For students who were members of the first course, the presentation is a review and a recommitment to the shared vision. For students who are returning after a brief absence from classes, the presentation is their orientation to the shared vision.

The focus of the second course is the development of the system documented by the menu skeleton. The class is divided into small programming teams of three students; each team is responsible for implementing the system. For individual accountability, each member of a team is assigned one entity class in the schema to implement (See Figure 4).

Comparison of 3GL Implementation Choices. The limitations of the sequential approach are examined during the study of indexed sequential and relative file organizations. In particular, the implementation of the project of the first course, sequential access of entity class D and table load/search, is compared to an implementation using sequential and random access of indexed sequential files. The relational operators of PROJECT, SELECT, and JOIN are examined in conjunction with the INPUT PROCEDURE option of COBOL's SORT.

The concepts of key, entity integrity, referential integrity, foreign key are revisited in light of the capabilities of the indexed sequential organization. To guide the students in completing the shared vision, it is sufficient to focus on one of the entity classes and the influence of associations between entity classes. The advantages of choosing indexed sequential file organization immediately become apparent when discussing the preservation of entity integrity and access by alternate keys. The impact of implementing referential integrity arises naturally in at least two circumstances:

during the analysis of requirements for entity class maintenance and during the analysis of report requirements.

For entity class maintenance, an ADD, a DELETE, and possibly an UPDATE are constrained by the need to preserve referential integrity. In particular, an ADD of a new entity in entity classes B, C or D (See Figure 2) or a change to a foreign key in those entity classes obligates the programmer to verify the existence of that foreign key in the appropriate entity class. Developing the application procedures to support this constraint reinforces the importance of keys and foreign keys. The impact upon data integrity by physically deleting entities in classes A, B, or C (See Figure 2) is likewise explored.

For reporting requirements, the impact of preserving referential integrity is explored in two types of reports. One type is the single-level control break for a 1-many association, e.g. from B to D (See Figure 2). This report is prepared by sequentially accessing entities from B and linking them to associated entities in D; this is done via an alternate key for the foreign key in D contributed from B. A second type of report that may be required is an ORPHAN REPORT, e.g. from D to B. This report is prepared by sequentially accessing entities from D and, using the value of the foreign key in D contributed from B, listing those entities of D which have no parents from B. This activity, perhaps more than any other, highlights the importance of referential integrity for the system being developed.

<u>Introducing Rudiments of the Relational, Hierarchical, and Network Database Models.</u> Discussing the relational data modelling approach is easily accomplished in this course. Using the schema from the shared vision, the relational data modelling approach is examined by comparing a sequential file implementation with an indexed sequential file implementation. The schema is again used as the basis for examining the hierarchical and network database models. This is accomplished by studying partially inverted and fully inverted lists.

<u>SQL, VIEWS, and Revisiting the Original Problem.</u> During the last week of the quarter, the rudiments of SQL are introduced to the class. Commands for creating TABLES and VIEWS and queries are presented. The principle of spiral learning is nicely accommodated when the original problem (Figure 1) is demonstrated to be a VIEW. An SQL query is developed for <u>assembling</u> the data needed for the project of the first course. A program employing an embedded SQL command is studied to see how that embedded SQL approach for assembling the data compares with the procedural capabilities of COBOL. By revisiting the original problem, students can see that the simplicity of access that was desirable for reporting can be achieved as originally envisioned (a single entity) without jeopardizing data integrity. The value of switching to a DBMS for both data maintenance and access is addressed. The use of the DBMS to verify program results from the first course is explain in terms of the flexibility of the 3GL (COBOL) for special data handling.

<u>Participatory Development in the Second Course.</u> Active student participation continues in the second course. The presentation of the shared vision is accomplished with assistance from returning students. The comparative analysis of sequential and indexed sequential implementations results in decisions regarding the implementation of entity integrity, object maintenance, referential integrity, as well as the types of reports to be included in the system.

Database Concepts in the Third Course. Because the University uses the quarter system, the Applications Development cluster extends through the first three weeks of a database management course. This third course concentrates on the development of more complex systems with a relational DBMS. The integration of database concepts and terminology in the third course consists of

- revisiting the shared vision,
- reusing the DBMS project of the first course in the first assignment,
- extending and completing the coverage of database concepts.

Revisiting the Shared Vision.
During the first three weeks, the shared vision is reviewed. Students who have completed the first two required courses will already be familiar with the schema and the menu system. As in the second course, this discussion will orient new students to the shared vision and the proposed direction.

Reusing the DBMS project of the first course
In the first course of the Applications Development cluster, a DBMS was utilized for two purposes: to implement the entity classes as tables and to verify the output for the first course's project. The entity classes created and the report prepared in the first course are used in the third course as a starting point for the implementation of the system using a DBMS. This spiral learning activity reinforces previously introduced concepts. Since the students are already familiar with the system requirements, they may concentrate on the tool rather than the problem. At the end of three weeks, students are required to present the completed system. The increased efficiency in development time through the use of the tool becomes evident to the student.

The Remainder of the Third Course.
Although it is part of cluster E, the remainder of the third course builds on the activities from the Applications Development cluster for the continued study of database concepts. The hierarchical, network, relational, and object-oriented data models for DBMS's are surveyed, normalization is presented and applied, and knowledge of SQL is extended. The third course culminates with a team oriented project in which the teams must:

- create their own problem statement,
- prepare a needs analysis,
- design a database solution,
- establish a plan for testing,
- implement and validate the solution,
- completely document the system, and
- make individual formal presentations.

A general table of contents for the final documentation is distributed as a guide for identifying high-level activities. All major phases of a team's project are reviewed and monitored by the instructor. A participatory development approach is again taken where the students are actively involved in the design and planning of their projects.

Conclusions

Summary. The Applications Development Cluster occurs between the Fundamental Concepts and Systems Development Clusters and is pivotal to the success of the model curriculum. By using a teaching method that emphasizes a database approach from the very beginning, students are able to reflect on the Fundamental Concepts Cluster from different perspectives. A rigorous treatment of database concepts in Cluster E - Systems Development, builds on the three different experiences, sequential files, indexed files, and DBMS.
By introducing database terminology and concepts in this manner, we not only comply with the recommendations of the IS'90 curriculum, but also help the students to understand a database perspective independent of a DBMS.

Benefits.
Some additional benefits of this approach are:

- Testing and validation are accentuated by the use of a DBMS in the first course.
- Frequently, the language of implementation can dominate rather than support the objectives of this cluster. By exploring issues such as the support of referential integrity between entities, the focus changes from the elements of the programming language to maintenance of data integrity.
- Similar applications can be introduced to reinforce database concepts and terminology and increase student understanding.
- Students are gradually introduced to team participation. The development of interpersonal skills necessary for team-oriented work will allow the project to be successfully completed on time. These skills are valuable for the remainder of the curriculum and in their professional career.
- Most importantly, when students actively participate in the development of the shared vision, they become more committed to its success. It follows that the students have a better overview of the relation between

applications and database concepts.

Observations.

Participatory Development of the Shared Vision: Merely distributing the schema and menu skeleton will not achieve a desirable result. Likewise lecturing about the schema and the menu skeleton will not result in effective communication with students. It is participation in the development of the shared vision that provides insight and, consequently, adds value to the shared vision.

Spiral Learning: Establishing and maintaining a shared vision provides an effective way to incorporate spiral learning in the three courses. Revisiting concepts revealed during the data analysis phase and reusing terminology throughout the three courses is a way of extending the cognitive level from Awareness to Literacy and perhaps even the Concept level [Longenecker and Feinstein, 1991]. Implementing the entire schema in the second course reinforces the concepts of entity class, entity, key, and associations among entities. Revisiting the schema in the third course provides the student with a third alternative for implementation.

Learning Levels: Table 2 summarizes the relationship between IS'90 and the approach described in this paper. The first two columns of Table 2 lists knowledge elements for database concepts from IS'90 and the exit level for cluster D. The third and fourth columns show our implementation and the corresponding exit level of instruction. The three courses discussed in this paper are labelled AD-1, AD-2, and AD-3; the prerequisite personal computer skills course is labeled PC-SKILLS.

The targeted learning level for instruction appears to exceed the level recommended in IS'90 when the described approach is used. However, this higher level does appear to exceed the abilities of the students.

Textbook Support: Only a few texts provide support for the integration of concepts into the Applications Development cluster.

REFERENCES

1. Bloom, Benjamin S. and Broder, Lois J., *Problem-solving Processes of College Students, An Exploratory Investigation.* The University of Chicago Press, Chicago, Illinois, 1950.
2. Bloom, Benjamin S., et al, *The Taxonomy of Educational Objectives: Classification of Educational Goals, Handbook 1: The Cognitive Domain.* McKay Press, New York, New York, 1956.
3. Covey, Stephen R., *Principle-Centered Leadership*, Simon and Schuster. 1991.
4. Daniels Jr., Robert M., Hayes, Glenda, Longenecker Jr., Herbert E., and Daigle, Roy J., "An Implementation of the DPMA IS '90 Curriculum for Information Systems". *Proceedings of the International Academy for Information Management*, (1992), 443-457.
5. Daigle, R. J. and Kemp, Janet J. "Managing Application Development - Introducing Project Management Into the Applications Development Cluster". *Proceedings of the Tenth Information Systems Education Conference*, (1993), 5-12.
6. Daigle, R. J., Doran, Michael V., Feinstein, David L., and Vest, Sharon N., "Perceptions of Faculty Evaluation: Consensus Based on Participatory Development", Proceedings of the 32nd Annual Association for Computing Machinery, Southeast Conference, p388-391, March 17-18, 1994, Tuscaloosa, Alabama.
7. Gronbaek, K. Grudin, J. Bodker, S., and Bannon, L., *Participatory Design: Principles and Practices*, Lawrence Erlbaum Associates, Inc. Publishers, Hillsdale, New Jersey, 1993.
8. Longenecker, Herbert E. and David L. Feinstein, Editors 1991c. "IS'90: The DPMA Model Curriculum for Information Systems for 4 Year Undergraduates", published by Data Processing Management Association, Chicago, 1991.
9. Longenecker, Herbert E., et al. "IS'93: The DPMA 2 Year Model Curriculum - A Natural Evolution From IS'90, A Strong Information Systems Foundation", *Proceedings of the International Academy for Information Management*, (1992), 253-260.
10. ACM (1991). ACM/IEEE-CS Joint Task Force. Computing Curricula 1991, February 1991.

Body of Knowledge Element	Exit Level
3.1.1 DBMS: Features, Functions, Architecture	1
3.1.2 Logical Design (DBMS independent design): ER, Object-oriented	1
3.1.3 Data Models: Relational, Hierarchical, Network	1
3.1.4 Normalization	1
3.1.5 Integrity: Referential, Data Item, Intra-relational	1
3.1.6 Data Definition Languages	1
3.1.7 Application Interface: DML, Query, SQL, etc.	1
3.1.8 Distributed Databases	1
3.1.12 Data Dictionary, Encyclopedia, Repository	1

Table 1

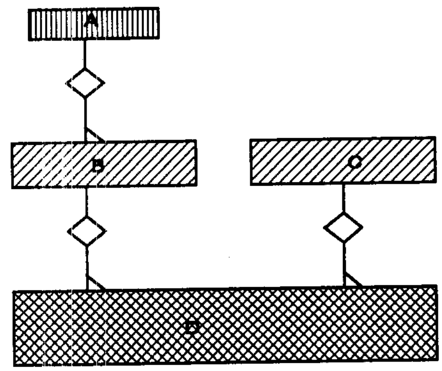


Figure 2

Body of Knowledge Element IS'90	Exit Level Cluster D	Courses	Exit Level of Instruction
3.1.1 DBMS: Features, Functions, Architecture	1	*PC-Skills* *AD-3*	*1* *3*
3.1.2 Logical Design (DBMS independent design): ER, Object-oriented	1	*AD-1* *AD-2* *AD-3*	*2* *2* *3*
3.1.3 Data Models: Relational, Hierarchical, Network	1	*AD-2* *AD-3*	*1* *2*
3.1.4 Normalization	1	*AD-3*	*3*
3.1.5 Integrity: Referential, Data Item, Intra-relational	1	*AD-1* *AD-2* *AD-3*	*1* *2* *3*
3.1.6 Data Definition Languages	1	*PC-Skills* *AD-1* *AD-3*	*1* *1* *2*
3.1.7 Application Interface: DML, Query, SQL, etc.	1	*AD-1* *AD-2* *AD-3*	*1* *2* *3*
3.1.8 Distributed Databases	1		
3.1.9 DBMS Products		*PC-Skills* *AD-1* *AD-3*	*1* *1* *2*
3.1.10 Database Machines			
3.1.11 Database Administration		*AD-3*	*1*
3.1.12 Data Dictionary, Encyclopedia, Repository	1	*AD-2* *AD-3*	*1* *2*

Table 2

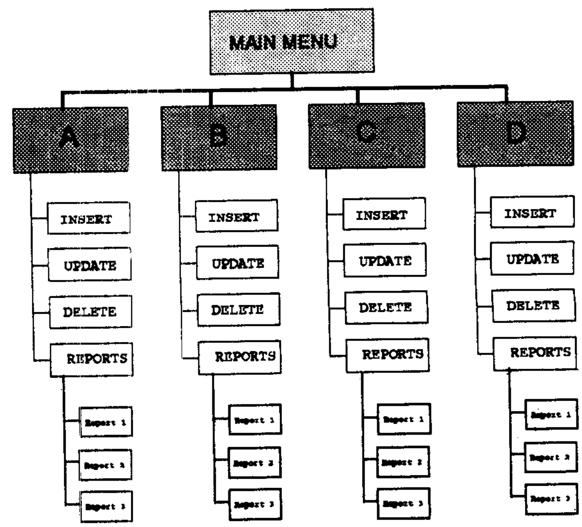


Figure 3

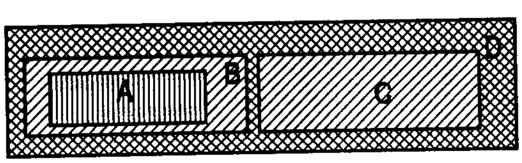


Figure 1

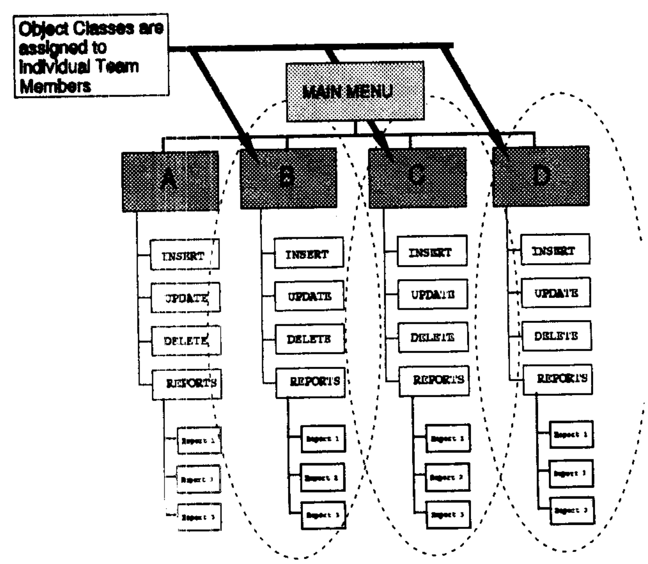


Figure 4

Affiliation With Industry: A Win-Win Relationship

John D. Haney
Computer Information Systems
Business Department
Walla Walla College
College Place, WA 99324
(509) 527-2018 hanejo@wwc.edu

ABSTRACT

An affiliation between a college and industry provides benefits to both the college and the associated institution. With an understanding that what is accomplished is within the framework of academic classes, then a win-win situation occurs. For the students, it is the opportunity to apply conceptual knowledge in a real setting. For the institution, the benefits of receiving computerized systems are significant.

INTRODUCTION

The needs of an industrial institution and the product of an information systems academic program are not always compatible. This reality is addressed in the curriculum goal of Data Processing Management Association Model Curricula for the 1990's. The primary goal of the model curriculum is "to facilitate graduating competent students in the field of Information Systems" (Longenecker, 1991).

The traditional academic framework does not always provided the necessary foundation to prepare students for performing sufficiently in the workplace. Eli Cohen has addressed an issue of concern regarding the continuance of old ideas. He feels that old "Idols" in education are obsolete and should be discarded. One of these idols is not discarding old material. New concepts are merely added to the old (Cohen, 1994).

Currently, an added dynamic is one of client-server development. The traditional approach in textbooks, on analysis and design and database development, is the systems development life cycle from the standpoint of large systems developed by the centralized computer center. Continuing to focus on the old has left a chasm in the curriculum.

One attempt to bridge the gap between what academic institutions produce and what industry demands is an affiliation between the academic institution and industry. This can be a positive step in moving toward achieving the goal of the DPMA model curriculum.

UNDERLYING PRINCIPLES

Problem solving is the foundation of the information system curriculum. The steps of the systems development life cycle of analysis, design, development, implementation, evaluation, and maintenance are the groundwork for this curriculum. Historically, the greatest area of focus in the curriculum, in a practical sense, has been in the development of computer programs. Although this is a vital component, it does not provide the student with a complete set of skills for solving business oriented computer problems.

The systems analysis and design, and database classes help address the overall scope of a system solution. But, particularly in the case of analysis and design, it remains to a great degree theoretical, or at best theoretical case studies. In an attempt to resolve this problem, and to provide a broader scope of skills for the students, an affiliation with industry is one possible alternative.

As part of the college's computer information system curriculum, an affiliation with the local Veterans Administration Hospital was established. This includes a formal agreement where both parties sign a contract. An understanding exists that the projects will be accomplished within the framework of academic classes. The interface with the V.A. Hospital is the computer department (Information Resource Management) which weighs the needs of the institution and recommends the projects that will be assigned to the college to work on.

The most appropriate classes in this setting are the systems analysis and design, and database classes. The analysis and design component comprises two quarters, with a focus on analysis during the first quarter and design the second quarter. The database course is offered the third quarter. This sequence allows the students to theoretically and practically work through the system development life cycle of analysis, design, development, and implementation.

This sequence of classes provides an entire year's framework concentrating on a common objective or related tasks. In this environment the students can take a problem that requires a computer solution and work through the entire systems development life cycle from conception to completion.

In the analysis class, the techniques of interviewing, observation, questionnaires, and data gathering are employed to gain first hand knowledge. This enables the students to evaluate how these differing techniques help to define the problem.

Instead of solving a predefined problem, the students, through the techniques of analysis, determine what the problems are. This helps them determine which techniques are the most beneficial in various situations because they are responsible for gathering "real" information.

In the design class, the greatest benefit has been learning the use of prototyping. The aspect of designing a system that will actually be used places a greater sense of responsibility on the students. This forces them to work closely with the users to design the system in such a way that it will be acceptable. The added benefit of this experience far exceeds the traditional design of case studies in a purely academic setting.

The database class provides the opportunity to fully develop and implement the system. Consequently, with the exception of evaluation and on-going maintenance, this three course sequence provides an environment for fully solving a problem that requires a computer solution by going through the entire systems development life cycle.

The type of projects currently incorporated in the these three classes were in the past one projects class. The results were often less than desirable because of limited class time. Therefore, the projects class was effectively blended into the analysis and design class, which was extended from one quarter to two, and the database class which is one quarter.

One of the projects selected for the 1993-1994 academic school year involved an inventory system for the computer department at the V.A. Hospital. This system keeps track of all the hardware, software, and users at the Hospital and logs all the occurrences of support by the computer department.

This project provided all of the elements of problem solving through the system development life cycle. Given the computer related nature of the problem, the students were able to understand the full scope of the issues involved.

COSTS - BENEFITS

The V.A. Hospital has a Client/Server network that provides basic information which previously resided on a mini computer. The users have access to the centralized database through terminals. The mechanism for accessing the data is cumbersome and the resulting format is awkward at best. During the past two years the number of personal computers has grown from four to over eighty. The primary demand of the users is to have the ability to download information from the centralized database to their personal computers, enabling them to manipulate their data using PC based databases and spreadsheets.

COSTS

Viewing this relationship from a cost-benefit analysis standpoint helps to pinpoint the true value of such an association. From the standpoint of the college it is a golden opportunity because of the "real life" lab that the V.A. Hospital provides. It also benefits the hospital, because the growth of personal computers has far exceeded the ability to provide support, due to budgetary constraints.

Within the hospital there are two components of consideration. The first is the computer department (IRM), the second is the end-users. The costs to the computer center primarily entail the time involved working with the students and faculty from the college. It is merely a matter of weighing the different needs of the various user departments and prioritizing the needs. Part of this cost is political, deciding who gets what and in which order. A more subtle cost is the potential transference of dependence upon the college for the solution to some of the users problems.

For the users the costs are time spent with the students, particularly in the analysis phase. A significant factor is placing on hold the needs of some users while others are being worked on. There is also a cost of security, as users are asked to change their old work habits and move toward processes that will enhance their activities in the long term but cost them discomfort in the short term.

The costs to the college mainly involve a change in methodology. For the faculty the classes are less structured, due to time constraints working with the personnel at the hospital and the various aspects of the projects involved. Blending the projects into more theoretical and case oriented classes places an administrative responsibility on the faculty and requires a balancing of theory and practice.

For the students the less structured format costs them time in establishing contact with the users and the personnel of the computer department. For the most part it is a one time experience, and the pressure of performing adequately is placed on the students. This creates an even greater potential cost, for both the college and the students, of reputation if the projects are not completed satisfactorily.

BENEFITS

The benefits to the V.A. Hospital are substantial. A major benefit for the user departments is support in the area of systems development that the computer center does not have the time to provide. With the large increase in personal computers over the past two years there has been expectancy placed on the users by the administration of the V.A. Hospital to utilize the computer resources. But the financial constraints have not allowed for adequate software support for the users. This is not only in the area of developing information systems, but providing education for some of the users.

The benefit to the computer department is reduced pressure to provide some of this service. The general position of the computer department, toward training users, is that if the users have reference manuals then they are responsible to train themselves. This has developed due to a lack of personnel but creates a high degree of frustration and pressure among the users. It has also created a position of resistance from some of the users. So the affiliation provides a benefit to the V.A. Hospital for both the end-users and for the computer department.

The benefits to the college are even more tangible. The students are able, through the systems analysis and design, and database classes, to apply theoretical knowledge to real situations in a time period that allows for the presentation of theory and application of knowledge in a real situation. This three course sequence provides a comprehensive framework in such a way that the entire systems development life cycle is incorporated into the development of the projects.

The benefit for the faculty is one of being able to present the conceptual framework, and then through practical experience receive feedback from the students in a relevant time period so that the concepts can be reinforced. This has been a tremendously rewarding experience.

The benefits for the students are multifaceted. There is the sense of accomplishment by performing tasks that are actually used, not merely academic exercises. There is also a solidifying of skills and techniques that they have learned which enables them to take a problem, that lends itself to a computer solution, and envision the process from start to finish.

The feedback from the students has been extremely positive. One student wrote,

> "Using the VA hospital as a project provided the perfect opportunity to apply class theory in a practical manner. It helped to reinforce my understanding of the subject matter" (Student Evaluations, 1994).

Another student's comment was similar,

> "The V.A. Hospital project gave me a chance to relate concepts learned in class in a real world business environment. I recommend that such projects be maintained in the class format" (Student Evaluations, 1994).

The feedback from the V.A. Hospital has been in a very positive vein with an expression of regard for the professional behavior of the students and the functionality of the projects.

SUMMARY

An affiliation between the college and industry, in this case a local Veterans Administration Hospital, provides an immeasurable benefit to both the college and the affiliated institution. If the affiliation is entered into with an understanding that the projects are being accomplished within the framework of academic classes, and realizing that any accomplishment is of value, then it is a win-win situation.

For the students, the opportunity to apply analysis, design, database, and programming concepts and techniques in a real setting provides them with theoretical knowledge and applied skills prepared to enter the work force.

For the affiliated institution, the benefits are the development of computerized systems that would not have been developed in the same time period.

The type of relationship between college and industry, that has been presented, can help move a Computer Information Systems program toward the goal of the DPMA model curriculum of facilitating graduating competent students in the field of Information Systems.

REFERENCES

Cohen, Eli Boyd. (1994). "Evolving Trends in Information Systems Education," Journal of Information Systems Education, 5(4), Winter 1993-94, pp 14-17.

Courtney, James F. Jr. and Paradice, David B. (1992). Database Systems for Management. Irwin.

Donlan, Hugh. (1994). "Model Curriculum Improves IS Education," Inside DPMA, November 1993, p 6.

Fertuck, Len. (1992). Systems Analysis and Design. Wm. C. Brown Publishers.

Gore, Marvin and Stubbe, John. (1994). Contemporary systems Analysis, Fifth Edition. B. & E. Tech.

Kendall, Kenneth E. and Kendall, Julie E. (1992). Systems Analysis and Design, Second Edition. Prentice Hall.

Longenecker, H.E., Jr. and D.L. Feinstein (eds). (1991). Information Systems: The DPMA Model Curriculum for a Four Year Undergraduate Degree, Data Processing Management Association-Education Foundation, Park Ridge, IL.

Rob, Peter and Coronel, Carlos. (1993). Database Systems. Wadsworth.

Stamper, David and Price Wilson. (1990). Database Design and Management. Mitchell, McGraw-Hill.

Student Evaluations. (1993-1994). Walla Walla College, College Place, WA 99324.

Whitten, Jeffrey L. and Bentley, Lonnie D. and Barlow, Victor M. (1994). Systems Analysis & Design Methods, Third Edition. Irwin.

Wu, Shih Yen and Wu, Margaret S. (1994). Systems Analysis and Design. West Publishing Company.

NETWORK DISTRIBUTIVITY:
An Alternative Solution to the Growth Problem

Dr. Longy O. Anyanwu
Department of Computer Information Systems
Georgia Southwestern College, Americus, Ga 31709
longy@gswrs6k1.gsw.peachnet.edu

ABSTRACT

Presently, network systems are the backbone of today's successful information processing and management. However, they have not been successful in providing for business-based organizational growth. A comparative analysis of available network computing architectures indicates ATM-based distributed intelligence system is the most plausible technological promise for tomorrow's business needs.

INTRODUCTION

Studies[1] have shown that as we entered into the 1970s and 1980s, the corporate value and demand for information processing grew[2]. There emerged a growing need to harness virtually all information resources with various data communication channels, resulting in the escalation of decentralized systems, often referred to as physical dispersal of computing resources[2]. Nelson and Patrick-Ezzell of AT&T noting the rather spectacular growth of connectivity in today's network computing, very succinctly stated:

> In 1990, an estimated 33 million PCs were in use in the U.S., and this number is projected to reach 69 million by 1994. Interestingly, the number of PCs connected to Local Area Networks (LANs) is growing at about 70% a year. LANs are growing at 20% a year. Some terminal equipments have growth rates of more than 100 % per year.[3]

Similarly, organizations naturally grow and many change business direction, and many more take on additional business functions. There are many problems created by increased information processing and dissemination. Perhaps, the greatest of these problems was the inability (possibly due to architectural inflexibility) of decentralized systems to provide for corporate growth and expansion, without excessive costs. For instance, it was not rare to see that many organizations abandoned their old equipments for new ones so that they might expand and integrate a range of growth-triggered new business functions, or even for efficiency.

THE GROWTH PROBLEM

Naturally, actual business operations may not be centralized, and usually are not. This is even truer now that organizational functions and functional mechanisms are becoming globally interconnected. It has become equally common place for business activities to be performed at geographically distant locations while their products are routed throughout the globe to various other locations for immediate utilization.

Organizations have wanted multi-user systems that will grow with the organizations and expand to multi-user multicenter (many users operating at more than one site at same time) technology with little or no performance detraction and minimal waste, if any[4]. Organizations and businesses want technology that is really affordable and upgradable with no or minimal obsolescence, even when organizational mission changes or new business functions emerge[5]. They have needed systems, that are easily customizable to their unique business needs and within their financial capabilities, systems with high tolerance to a variety of data forms (including voice, video, fax, etc.), systems with high throughput capability to accommodate anticipated surging volume of communication traffic with very minimal bottlenecks[6].

MAJOR WEAKNESSES OF CENTRALIZED AND DECENTRALIZED SYSTEMS

Both centralized and decentralized approaches to information processing are structurally insensitive to organizational growth, and in many cases, have actually inhibited expansion and stifled progress[7]. Information update has not been prompt, owing in part to the batch-update orientation, and delay times in the transportation of update data to central site, *id ipse*, was an exposure of update data to security or safety breach. Besides, systems professional activities are often bogged down with the rampant political battles over administrative turf inherent with the power of centralized information processing. Uncontrolled escalation of on-line access to centralized processing frequently results in sluggish system performance and this is even worse at work peak times, when processing speed is of critical importance. This weakness undermines business operations and often precipitates loss of corporate competitive advantage. Indubitably, this dangerous loss of competitive advantage may mean the eventual downfall of an organization. Nevertheless, as work volume increases to trigger bottleneck and associated problems, and operational response time becomes crucial to successful business functions, it becomes the more evident that mere distributive network systems could not be a total solution, therefore triggering off the exploration of distributed intelligence as an option.

THE SALIENT PROMISES OF DISTRIBUTED INTELLIGENCE

Nevertheless, have distributed intelligence systems (integration of distributed databases, distributed network and communication systems) become the hope for tomorrow's information processing, distribution, management and use? Is DISTRIBUTIVITY a single answer to myriad of problems? Of course not. Invariably, distributed intelligence, without necessary communications and network infrastructural underpinnings, would actually mean little. Communications and network architectures and their configurations are essential to the successful implementation of a distributed intelligence system. Notably, many network configurations have been explored over the years, and their relative strengths and weaknesses have also been noted.[8] Various network topologies are descriptively representative of the different degrees of distribution of information resource and processing needed to balance both organizational and processing effectiveness and efficiency[9]. Many commonly used distributive network topologies have been a significant part of the progressive solution to this complex problem, namely:[10] Vertically Distributed Network System (VDNS), with a central control and remote nodes as implemented in many multi-site supermarket stores; Semi-autonomous Distributed Network System (SADNS), with a main control processor and several multi-center control processors as typically implemented in national organizations with regional distribution or coordination centers; Satellite Distributed Network System (SDNS), which actually may be an example of the SADNS except that no processor is identified as central or controller; Autonomous Local Area Network System (ALANS) or Stand-Alone LAN system, with shared resources as may be implemented in a business department, where a node may be intelligent(stand alone), but share resources with other nodes; Horizontally Distributed Network System (HDNS), sometimes referred to as Load Sharing Network System(LSNS)[11], consisting of a network of SADNSs, where overloaded processors shift tasks to other processors, and there is no central controller, thus the system is not crippled at the failure of a node of systems and according to need, a system may be activated or shut down. It must, however, be borne in mind that many corporate network systems evolved, and do evolve, in stages, usually from both bottom-up (as need for network systems arises at operator level, in normal execution of organizational functions) and top-down (top management for strategic considerations utilize network systems) perspectives. In such a case, as the corporate network topological paradigm emerges, distributed communication controllers, file servers, database machines, etc. are installed as needed or as available at department or other levels. Would, therefore, the HDNS be the way to go to implement efficient distributivity? Perhaps. The performance of a distributed intelligence is obviously a function of both the network configuration and the capabilities of the processor architecture type, which is a major determinant of the nature of processor activity, including effective interprocessor communication (IPC). Many basic architectures have a varying range of processor interactional performance in accordance with the design of the processor coupling (tight, where processors share common memory or loose, where processors interact through communication lines). Some of these basic architectures include: Single Instruction Single Data Stream (SISD), as implemented in sequential instruction processing machines; Single Instruction Multiple Data Streams (SIMD), as in array processors where a single instruction is simultaneously applied to all elements of the array; Multiple Instruction Single Data Stream (MISD), as in pipeline processing, where a processor may concurrently be executing several instructions (a kind of sequential execution of chunks of several instructions at the same time), a technique often typified in time-share systems; and Multiple Instruction Multiple Data Streams (MIMD), as in multiprocessor systems, where different processors can take on different portions of a problem. The MIMD architecture naturally involves a high degree of both static and dynamic IPC. Notably, the degree of this processor interaction may be at different operational levels, namely: job level, where jobs are exchanged among loosely coupled processors; task level, where programs/subprograms are exchanged among processors; process level, where separate program modules may be concurrently executed by different processors; and instruction level, where concurrent processing of multiple instructions is usually performed by tightly coupled processors. However, a fundamental question must be answered, namely, which processor architecture is best suited to the network and communications infrastructure of tomorrow, or vice versa?

The Concurrency Factor

The benefits of concurrent activities of multiple processors in distributivity cannot be overemphasized. According to Amdahl's (processing speed factor) law, $S_{max}=P/(fP+1-f)$, (where S is the speed factor, P is the number of processors involved in a task, and f is the number of parallel operations composing the task), multiple processors involved in a distributed intelligence or even in a distributed network system may work concurrently at different tasks within same problem (parallel processing) or on different but related problems (distributed processing). Either way, the purpose of throughput maximization and efficient utilization of system resources is ultimately realized. And good throughput has been realized for a number of distributed intelligence systems. However, have the growth concerns of the business community been addressed adequately? Certainly, not.

ALTERNATIVE SOLUTION TO THE GROWTH PROBLEM

The growth problem is a dynamic one, and any static solution is no solution at all. Static or proximate solution to it today is, at its best, only good today and may fail woefully, as it has hitherto, in the face of tomorrow's business innovations, technological surprises and new social tenabilities. However, a stable solution frame with a very high degree of flexibility and sensitivity to the individuality of organizational character and function, a frame with reliability, easy pliability and high tolerance level to a wide range of variables, will enable quick and smooth customization to organizational unique carriage to the growth issue. Many computer-based technology experts agree that the *sine qua non* ingredients to the construction of such a solution technology will include the following features[12,13].

Modularity
Modularity is as essential as it is fundamental in the design, manufacture and configuration of computing, network and communications resource architectures. Modularity is indispensable in the development and implementation of software[14], network infrastructure, communications strategies, organizational structure and functions and business techniques to enable smooth expansion and enhance organizational business growth.

Network Portability
The network component constituency must be portable and affordable enough for easy assemblage and adaption to organizational needs. Three important integral characteristics of this portability are the horizontal compatibility of network components, the scalability of the network configurational architecture to suit particular tasks, needs and price tags, and the capability of the distributivity to be application driven for easy customization and management[15].

Vertical Compatibility
It must be understood that the issue of abandoning the old or traditional architecture and hardware for the new, or the cost of converting hardware and software to new network topological underpinnings is very scary to business executives. Therefore the level of compatibility must ensure smooth and efficient transition with minimal difficulty, complexity or waste.

System Reliability
The self-reliance of the system must stand the test of time. During the later 1990s' projected shortage of technically trained maintenance personnel, the expected few maintenance personnel will either be too busy keeping complex distributed intelligence systems operational or too expensive for small businesses to compensate. Thus, good systems will be expected to be self-maintainable.

High System Throughput
With projected improvements in data transfer rates of communications infrastructure and a high degree of IPC, the system throughput must be maximal and capable of weathering the expected heavy traffic of the explosive escalation of network computing. As a growing number of corporate executives acknowledge the value of information, and as information value for business functions rapidly increases, the indispensability of information in corporate strategic advantage crystallizes clearer. This means more demand for information and therefore a boom for the computing and network industries. Undoubtedly, tomorrow's rather elaborate information manipulation and transfer, which visibly will include the integration of data, voice, video, imaging, graphics, dial-up, e-mail and video-conferencing network systems will need the technology that assures the capability for sufficient robustness, speed and flexibility to effectively handle the heavy and congested network traffic of myriad data packets simultaneously travelling more or less at the speed of light in virtually all directions in the network. The best the current or near future technology can offer is, perhaps, the Asynchronous Transfer Mode (ATM) infrastructure, a switched circuit packet technology with higher speed than other technologies such as Frame Relay or x.25. Figure 1 highlights relative capacities of major network technologies. A rather quick survey of current and near-future technologies for versatile high speed and high capacity distributed intelligence infrastructure including the Experimental

University Network (XUNET), described as tomorrow's cell relay technology utilizing the ATM; the InterSpan Information Access Service, an x.25-based or Frame Relay-based service;

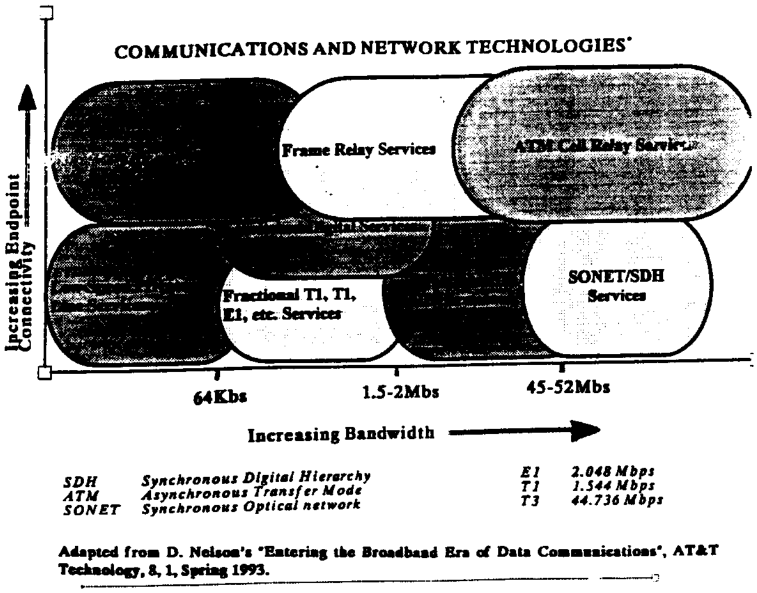


FIGURE 1

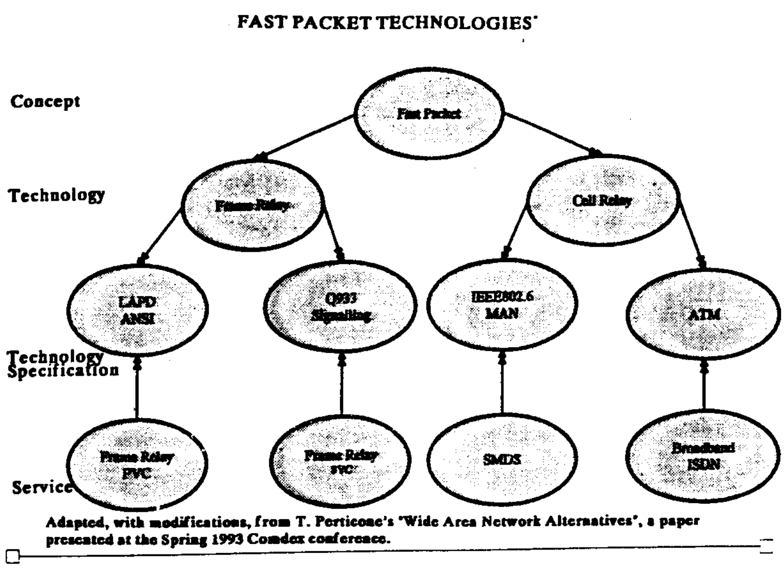


FIGURE 2

the Synchronous Optical Network (SONET), a U.S. version of the emerging international signal format for optical fiber networks with projected transmission rate of 2.4 gigabits per second, well above today's 45Mbps; and Mobile Remote Network Service, projected to be high speed multimedia transmission-based reveals that ATM has a surpassing capability- and reliability-based promise for the future, particularly as many-to-many endpoint connectivity increases.

ATM'S Technological Edge

While most other technologies are frame relay based on either Permanent Virtual Circuit (PVC) or Switched Virtual Circuit (SVC)[16], ATM is an B-ISDN-based cell relay packet technology. Figure 2 shows the architectural bases of the major technologies. Some fundamental advantages for the ATM infrastructure will include the following.

High Capacity and High Speed Technology

The ATM is a broad bandwidth and high speed technology which is capable of providing a projected load capacity transmission rate of 2.4 gigabits per second with a comparative minimum delay time (over 40 times less than the delay time for current AT&T's T1 technology), thus, drastically reducing the cycle time. The SUN Microsystems Corporation eloquently states:

> ATM has no speed or throughput barriers.... ATM offers a deterministic quality of service or latency. This means that videoconferencing application can be guaranteed of smoothness in delivery to the receiver. ATM is unique in this ability among the LAN technologies.[17]

The robustness of an ATM-based system to function even more comfortably as the many-to-many end-point connectivity grows (see figure 1) accentuates the usefulness of ATM technology in future distributed intelligence systems. Some experts postulate that the MIMD architecture is better suited to highly efficient distributed intelligence system implementation mainly for its advanced capabilities of both static and dynamic IPC. The transmission speed and capacity, fault tolerance, load shareability, relative autonomy and architectural modularity of an ATM-based distributed intelligence system appear to excite many.

Multimedia Capacity

The ATM's cell switching and relay base enables the switching and transport for multiple services ranging from connection-oriented to connectionless networks, from isochronous to multicolor services, from uni-medium to multi-media integration services including voice, video, fax, imaging, graphics, e-mail, videoconferencing networks and data[18]. Table 1 highlights the major fast packet technologies and their characteristics. Also see figure 3 for comparative bandwidth requirements of the various information media/services. Researchers will be able to interact easily among themselves and new products can be brought to the market more quickly.

COMPARISON OF THREE MAJOR FAST PACKET TECHNOLOGIES

CRITERIA	FRAME RELAY	SMDS*	ATM
Access Speed (bps)	Defined at T1 (46K-1.5M) rates, may reach T3 1.5-45M	Defined at T1-T3 (45M-622M)	Defined at T45 & more
Base Technology	Permanent Virtual Circuit(PVC) to Switched Virtl. Circuit(SVC)	IEEE802.6 Metropolitan Area Network (MAN)	Broadband ISDN
Application	Data	Data	Voice, Video, Data, etc.
Switching Tech.	Frame-oriented	Cell-oriented	Cell-oriented
Transport Unit	Variable length frame (less than 4096 bytes)	Fixed length cell (53 bytes)	Fixed Length Cell (53 bytes)
Transport Service	Connection-oriented	Connectionless	Connection-oriented and/or connectionless
Protocol	Thru layer 2	(Thru layer 2)	Supports higher layers in user, control & mgt. planes
Primary Thrust	IXCs, RBOCs	RBOCs	CPE, RBOCs, IXCs
Availability	1992-1993	1992-1993	1993-1994

Switched MultiMegabit Data Service
Data based on manufacturer information on the packet technologies.

TABLE 1

Business communication will become more or less instant and distance-based processing time lags will be imperceptible to the user. Organizational consumption of externally produced information elements will grow by leaps and bounds.

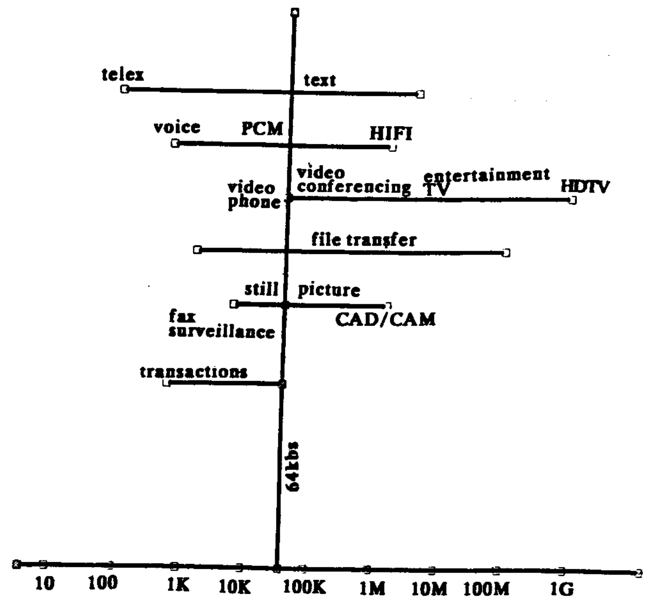
Great Flexibility

The capability and robustness to effectively consolidate a myriad of networks and architectures surely keep the ATM on the forefront of other technologies. Its management capabilities and increased responsiveness are already thrilling to potential users. The obsolescence issue will have been significantly addressed.

Scalability

Owing to its primitive level modularity, the infrastructure can be easily and

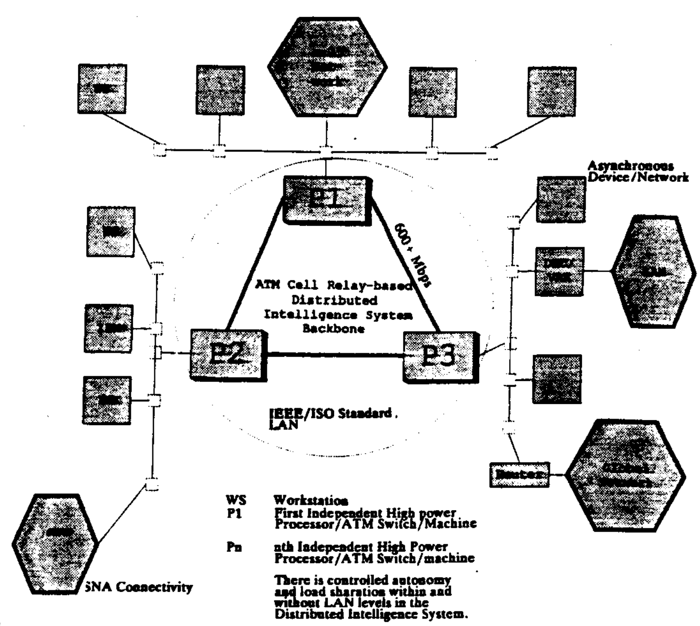
INFORMATION MEDIA BANDWIDTH REQUIREMENTS



Relative bandwidth requirements of various media, arranged for comparative purposes

FIGURE 3

CONCEPTUAL ATM-BASED DISTRIBUTED INTELLIGENCE



Design based upon processor architectural strengths and optimized capabilities.

FIGURE 4

selectively scaled up to include extra components/nodes and provide room for expansion and growth, or down to exclude system components/nodes to cut costs while adequately meeting user needs.

This is, perhaps, the single most important advantage to many small businesses. Figure 4 demonstrates typical scalable global, WAN or LAN network topology in an ATM-based distributed intelligence system, where the nodes may be major processors (for large systems) or terminals or even non-existent (for some small businesses). Virtually any configuration is possible.

Application Driven

Notably, the ATM has a robust addressing capability that enables easy and smooth access to applications and rather excellent interoperability between LANS and/or applications. Processor distribution and configuration can be neatly packaged from the application level. New applications can be developed to support evolving and emerging business needs. A comparative simulation of five frequently used network applications[19], shows that Netware 3.11 has a slight edge over the others. See table 2.

NETWORK MGT OF SELECTED SYSTEMS COMPARED

Criteria	Wt	Netware 3.11	Alertview 1.0	Visnet 2.0	Lanlord 1.0 beta	Snaprobe 1.0
Install/Doc	1	Satisfactory	Satisfactory	Excel.	Good	Good
Ease of Use	2	Good	Good	Good	Good	Good
Flexibility	2	Excellent	Good	Good	Good	Excellent
Interoperable	2	Excellent	Good	Good	Satisf.	Good
Alert Handle	2	Good	Good	Good	Satisf.	Good
Total	9	30	26	28	23	29

Software rated on a scale of Unacceptable(0), Poor(1), Satisfactory(2), Good(3) and Excellent(4), based on comparative operational simulation.

MEANING:

Poor -falls short in essential areas.
Satisf. -meets essential criteria and performs at basic expectations.
Good -meets all standards and has additional useful features.
Excel. -exceeds required and expected standards and has extra useful features.

Adapted with expansions from Eric Harper's 'Windows-based Network Management Comparison Master Card", LAN TIMES magazine, New York: McGraw-Hill, July 20, 1992.

TABLE 2

It must, however, be borne in mind that these and other emerging network applications still need a lot of refinement work to be fully adaptable to meet the future needs and fully utilize the capabilities of the described ATM technology. ATM's real-time capability and very low latency are remarkable.

CONCLUSION

Although ATM technology is far from being a perfect solution to all of our perennial technological problems, and as a matter of fact, ATM has its own inherent limitations and disadvantages (such as the ATM's T45 access speed considered slow for the anticipated load of the future information highway, and the fact that the ATM's technology has not been, and may not be fully realized in the near future), yet it must be admitted that the ATM technology today offers us the best and realistic alternative solution to most of both today's information integration and distribution limitations and those of tomorrow's foreseeable needs. At the same time, the successful implementation of this information integration, management and distribution very much depends on an effective concertion of thoughts from all concerned at the various hierarchical levels of information processing and use. In agreement, Jayne Fitzgerald of AT&T's ATM architecture center adds: "Such a collaboration (of thoughts) is necessary to develop strategies for the convergence of voice, data and video during this decade"[18].

SOME NEEDED STUDIES

Even with this consolidation of ideas, the technological transition to the ATM-based distributed intelligence system will no doubt pose a set of dilemmas. Perhaps, in the mind of the business or corporate executive, the foremost two of these questions will include: 1) What will happen to the existing architectures that were neither designed and manufactured for ATM compatibility nor for multimedia vertical compatibility? Will they all (or some) be re-engineered for continued use or abandoned to make way for a more versatile ATM-based architecture? What is the price tag? 2) At what level in the information processing, distribution, management and use should the multimedia integration first occur? Should it first occur at the indirect user level, where information interpretations and use are made, at the direct user level where information is produced at the desk-top or at the ATM hub? These areas of genuine concern need be further studied and addressed.

BIBLIOGRAPHICAL REFERENCES

1. William E. Leigh and Clifford Burgess, Distributed Intelligence: Trade-offs and Decision for Computer Information Systems, Cincinnati: Southwestern Publishing Co., 1987.
2. Raymond McLeod, Jr., Management Information Systems: A Study of Computer-based Information Systems, New York: Macmillan Publishing Co., 1993.
3. Nelson David E. and JoAnn Patrick-Ezzell, "Entering the Broadband era of Data Communications", AT&T Technology: Products, Systems and Services, 8 1, Spring 1993, p.2.
4. Debra Neidermiller-Chaffins, Inside Novell Netware, Carmel, Indiana: New Riders Publishers, 1992.
5. Philip Laplante, Real-Time Systems Design and Analysis: An Engineer's handbook, New York: IEEE Computer Society Press, 1992.
6. Carroll W. Frenzel, Management of Information Technology, Boston: Boyd & Fraser Publishing Co., 1992.
7. Kenneth Laudon and Jane Laudon, Management Information Systems: Organization and Technology, New York: Macmillan Publishing Co., 1994, 678-695.
8. Thomas Casavant and Mukesh Singhal, Distributed Computing Systems, Washington: IEEE Computer Society Press, 1994.
9. Thomas W. Madron, Local Area Networks: The Next Generation, New York: John Wiley and Sons Inc., 1990.
10. John M. Griffiths, ISDN Explained: Worldwide Network and Applications Technology, New York: John Wiley and Sons Inc., 1992.
11. Paul L. Tom, Managing Information as a Corporate Resource, New York: Harper Collins Publishers, 1991.
12. John McGregor and Arthur M. Riehl, "The Future of High Performance Computers in Science and Engineering", Communications of the ACM, 32 9, New York: ACM Press, September 1989, 1091-1101.
13. Tamer Ozsu and Patrick Valduriez, "Distributed Data Management: Unsolved Problems and New Issues", Distributed Computing Systems, Los Almitos, CA: IEEE Computer Society Press, 1994, 512-544.
14. Jeffrey L. Whitten, et al, Systems Analysis and Design Methods, Boston: Irwin Publishers, 1989.
15. Craig Hunt, TCP/IP: Network Administration, Ca: O'Reilly & Associates, Inc., 1992.
16. Tony Perticone, "Wide Area Network Alternatives", (paper presented at Spring 1993 Comdex International Computer Conference), 1993.
17. SUN Microsystems Computer Corporation, LANs of the Future, March, 1993.
18. Jayne Fitzgerald, "ATM: Vision of the Future", AT&T Technology, 8 1, Spring 1993.
19. Eric Harper, "Comparisons", LAN Times, New York: McGraw-Hill, Inc., July 20, 1992.

THE HUGE LEAP IN THE USE OF EMERGING IS-RELATED TECHNOLOGIES

Herman P. Hoplin
School of Management
Syracuse University

ABSTRACT

In a rapidly changing environment fueled by competition such as we have today, the **ISECON** theme--"IS Education: For Today and the Next Millennia," may appear to be somewhat elusive but not unreachable. In accepting the challenge it is of prime importance that we get a jump-start on emerging technologies.

INTRODUCTION

Much can be learned from IS's brief history. We have been slow to grasp trends on a timely basis, such as downsizing to microcomputers, using modern database management systems, going toward end-user computing, the use of 4th generation languages, networking with client/servers, integrating IS with business goals, etc. So today we face important critical and organizational issues, architectures, methodologies, implications and problems which are of primary concern to managers today. It is now up to information managers and users to determine how these factors focus on optimizing organizational success.

Discussions at recent conferences across America have seriously questioned the role and impacts of IS on organizations. Much of this problem was brought about by a recalcitrant attitude in IS shops that fail to realize the needs of the business. This was brought about partially by IS failure to grasp the need to act rather than re-act after it was too late. Some of this blame rests with business schools who have not been teaching information technology (IT) with business needs in mind. Collectively, the criticism of IT indicates that now that we have received the "wake-up call" it is high time that our major emphasis should no longer be IT *per se* but rather, it must be IT-supported business processes. In recognizing the huge leap in the competitive use of IT, Steven Alter maintains that although the focus remains on the computer, the business value is on "IT-supported business processes.[1] This point is illustrated in Figure 1. Also magazines such as the *CIO*[2] have had cover articles on the idea that, "When IS management meets with business management, the sparks can really fly and the results can sometimes be spectacular." The point here is that when business and IS executives work together they can devise an IT plan that solves critical business problems. This conference paper focuses on recognizing and exploring trends of several major emerging IS-related technologies.

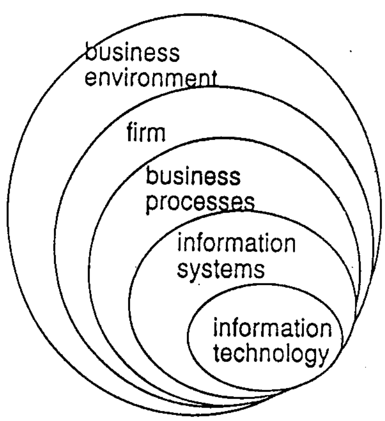


Figure 1. Levels for Understanding IT

MAJOR EMERGING TECHNOLOGIES

The rapid proliferation of emerging IT 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, data management, spread sheets, and data communications are changing at a greater pace than the current IS organizations can absorb.

The critical areas of the mid-1990's are heaping more coal on the fire of emerging technologies such as:

1. Distributed Processing including data communications and networking.
2. Windows and Client/Servers.
3. Modern Database Management Systems.
4. Impacts of Stakeholders and End-User Computing.
5. Bringing Languages into the 21st. Century.
6. Use of Automated Tools (CASE).
7. The Data Highway.
8. Expert Systems, Neural Systems, and Virtual Reality.

⑨ Electronic Data Interchange (EDI).
⑩ Other Types of Connectivity yet to be developed.

The above list was confined to ten critical trends now or about to get underway. If there ever was a condition of giving a drowning man a drink of water, IS faces that condition now. A brief synopsis of these critical and trendy developments follows:

<u>Distributed Processing including data communications and networking</u>--Foundation developments in this area triggered many of the other nine critical trends summarized in this section of the paper. The word "distributed" carries with it a plethora of massive developments. It broke open the need to break computing out of the black box in corporate headquarters. The "central" tendency to do computing in one place was reversed to bring data processing nearer to where the work was actually being done.

As organizations became larger and spread out into an expanded environment, it made sense (although the need was not recognized at the time) that computer processing had to be distributed. An early attempt was made to distribute the database to meet organizational demands. This effort was thwarted initially by technical problems that threatened the integrity and security of the database. Before this could be resolved the microcomputer onslaught occurred which not only effectively forced distributed processing but slowed down other developments while end-user computing became common-place and mainframe professionals began to feel threatened.

In the midst of the turmoil mentioned above, data communications became the pawns of networks, both local and wide. This had wide-spread organizational implications in that it forced a merger of sorts on data processing and data communications which took on International or Global ramifications which involved both national and international standards organizations. The good old days of data processing are gone forever and the management challenges of this are of no mean proportions. This leads us to the next trendy development which could be referred to as, 'Windows and Mice and everything nice.'

<u>Windows and Client/Servers</u>--No CIO or IS manager can afford not to look into these developments. Because of robust vendor influence great care must be exercised before evaluating the situation and taking the plunge. Windows no doubt are an advance in the technology but they may not be the silver bullet they are cracked up to be. Ideal systems should permit using either the DOS system or a windows application when it is advantageous to graphics user interfaces (GUI); however, for other transactions windows may just slow down the process.

Client/Server technology is now permeating the industry and it comes in a number of forms. Client server implementations usually appear in the form of client servers, file servers, and/or database servers or combinations of these in order to distribute applications across networks. The processes (clients) make requests to the servers through a procedure that allows a client on one machine to initiate a function on another machine which returns the results to the client's machine as if the transaction has been performed on the originator's machine. The server approach offers an attractive alternative to traditional computing in that it offers an immediate user payoff which "... includes empowerment via access, user participation, and productivity impact.[3] As indicated in Figure 2, a system can be either a client or a server or both.

<u>Modern Database Management Systems</u>--This is a must for effective management of both the large and complex data complexes and the small and simpler highly competitive data banks existing in the world today. Newer data models such as Relational and Object-Oriented systems are rapidly replacing the legacy systems of Hierarchical and Network. With data processing and data communications making quantum jumps in speed, distribution, and networking, it is essential that data junctions do not become clogged and bring a paralysis to vital processing and communications at the expense of stakeholders both within and external to private and public domains. The best planning, analyzing, and designing will be needed to further extend database technology.

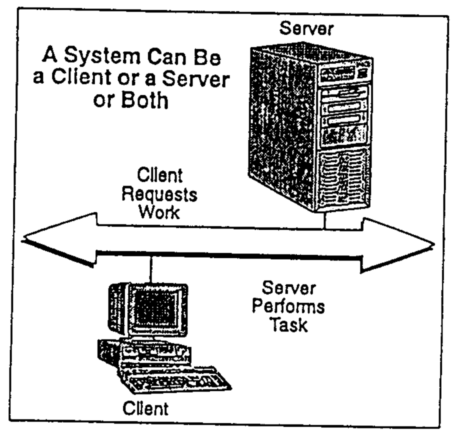


Figure 2. Eric Byers Schematic

<u>Impacts of Stakeholders and End-User Computing</u>--The human involvement and impacts of stakeholders in

organizational and applications development is not overlooked in this presentation. This also includes the user dimension and the relations between users and specialists. This is particularly important in the movement of the human-computer boundary where the user is becoming much more involved in the process.

Note is also taken of the increasing pressure and need for the integration of systems which are not being drawn together to integrate MIS technology in the organization of the future. Whereas early developments were focused on programming and data processing with a preoccupation of getting the computer to work, a significant shift has now occurred with the emphasis on the applications that the computer systems can do to help the business succeed in a highly competitive environment. This requirement makes it all the more important for MIS to succeed since it is now vital to business success. In fact, some authors have found it useful to consider MIS as a business within a business.

Bringing Languages into the 21st Century--Because of the dramatic shift to the PC environment and because of the failure to make quantum jump improvements in the productivity of 3rd generation (procedural) language programmers, it is necessary that more adjustments be made toward 4th generation languages (4GL's), database access languages (SQL), and other natural-type languages. It took about 30 years to progress from COBOL to "C" programming, a procedural language now greatly used with Windows, Expert Systems, etc. James Martin observed in the mid-1980's that for a quarter of a century the languages for commercial computing have evolved only slowly and that with 4GL's computer power becomes accessible to non-technical people without needing extensive training in DP.[4] Martin also stated that it is highly desirable that users who cannot program should be able to query the data base, extract from them the information they need, and generate reports, and in some cases update the data.[5] Many 4GL products are now available such as EXCEL, FOCUS, RAMIS, LOTUS to name several.

Use of Automated Tools (CASE)--In the 1990's we need to expand into tools for the Information Age and take advantage of the use of automation for planning, analysis, and design of application systems. A bi-product of this is greatly reduced maintenance costs principally in the area of the use of personnel for something that the machine can do. One of the better documented areas in this rapid trend is the marketing and use of Computer-Aided Software Engineering (CASE) tools. CASE has now become vital in both large and small projects. It is now a technology for all software developers. This is now seen as a great productivity increase for current and future software engineers. The presentation includes the target resource profile of CASE as well as its benefits, organizational changes needed, and some examples of vendors and implemented systems.

The Data Highway--Although scarcely more than a dream and a buzz word at the present time, the Data Highway is under construction and the Federal government has already funded a number of pilot projects. It involves an information infrastructure of telephone, cable, and internet systems and could be described as the logical conclusion of today's convergence of hardware, software, and networking technologies brought about by the increasing digitalization of data according to Oracle's Abrahamson.[6] This development will have a significant effect on business and society and it behooves IS planners and managers to have strategies down the road to utilize something that will be bigger than telephones and TV's as we see them today.

Expert Systems, Neural Systems, and Virtual Reality--Of the ten critical and trendy developments addressed in this paper, this one is the most tenuous at the moment. It is almost entirely in the realm of **Artificial Intelligence** which is a sensitive area in the business environment. There are a number of Expert Systems (ES) in the business world spanning applications in management, finance, manufacturing, medicine, and consumer products. Some research has been done in blending database and ES technologies into what one might call "Smart" systems. This critical area encompasses knowledge-based systems, computer vision, speech technology, robotics, fuzzy systems, neural networks, and evolutionary computing, knowledge engineering systems, and case-based reasoning, The March 1994 issue of *COMMUNICATIONS of the ACM* focuses on Artificial Intelligence and is an excellent up-to-date reference.[7]

Electronic Data Interchange (EDI)--This process refers to companies exchanging information electronically instead of using paper. It is a computer-to-computer communication of any machine-readable medium exchanged between trading partners. It is usually thought of as direct data communications linkage between respective computers. Direct data lines are either leased or dial-up. EDI is an emerging business practice that has the potential to affect the way organizations communicate common interbusiness functions.[8] The main purpose of EDI is to provide a communication standard for the electronic transfer of business transactions. Much needed action has been taken in EDI standards on the retail side of the industry (e.g. ANSI X12). Except in certain specialized areas EDI has not progressed as rapidly as one might have expected, primarily due to lack of standards, legal signature problems, external auditor questions, and security. However, because of progress in ironing out the bugs, feeling in the industry is that EDI is poised for take-off in and between companies and corporations, both large and small. When the cost-effectiveness of EDI becomes apparent and government controls are standardized there is little doubt that EDI will become a standard business practice prior to the turn of the century.

Other Types of Connectivity yet to be developed--With many areas of research underway it is more than likely that breakthroughs will occur that will update this list of state-of-the-art developments and will again change the complexion of IT in organizations. Parallel processing, space technology, new living systems applications, multimedia innovations, reengineering of computer-based systems, genetic algorithms, and platforms in communications hitherto not envisioned are only a few of the possibilities for the next century. The point here is that IS managers must have a strategy that will quickly sense a revolutionary trend in planning for the future if their company is to thrive in tomorrow's environment.

MAXIMIZING METHODOLOGIES AND RESOLVING ORGANIZATIONAL ISSUES

Methodologies operate within the context of an organization. In order to determine the appropriate methodology, we must consider certain organizational issues. These consist of the organizational structure, culture (power and politics in the organization), and the organizational position of Information Systems (IS).

Organizations can be categorized by their structure: functional, divisional, matrix, centralized and decentralized.[9] This structure will determine the official lines of control in the organization. This will, in turn, determine the type of technology to use and its appropriate implementation strategy. IS systems professionals need a full understanding of the structure of the firm in which they operate. They must also have an understanding of how certain technologies and implementation strategies can best fit that organizational structure.

A frequently used methodology in systems development is the Critical Success Factor (CSF) method. "The CSF methodology is a procedure that attempts to make explicit those few key areas that dictate managerial or organizational success."[10] It requires that the analyst or consultant have a strong understanding of the business. That understanding is then used to elicit information from executives regarding critical success factors for the organization. These CSF's can be used to promote organizational redesign issues and initiate strategic planning. Simultaneously, CSF's can be used to define information technology capabilities for the organization and eventually link the organizational and information needs.

This methodology encourages communication throughout the organization. Rockart cautions that this process will not work at all times in all companies. Timing is the key. Management must be ready to be involved. Competitive pressures or sheer awareness of the increasing strategic importance of information systems are among a long list of factors on which the outcome of the process is dependent.[11]

CSF methodology has a place in information systems development. Depending on the organization, it can have great strategic implications. The eliciting of critical success factors serves two purposes. The first to help management identify those issues that are increasingly important to the organization. Second, is that it gets management actively involved. It is not only a system development effort; rather, it is an organizational development technique. It has far reaching consequences for integrating organizational and IS goals.

Other potentially valuable methodologies are: Ends/Means Analysis (MIS Research Center, University of Minnesota); Business Systems Planning (BSP--IBM); Nolan's Stage Analysis; and Andersen Consulting--M-1.

Communications also play a role in organizational IS. A model for evaluating communication patterns in an organization as a means of determining the current organizational structure was developed.[12] In this model, they examined the link between the structure of the organization and the information technology being used. The outcome identified the implicit structure, or internal power in the organization. Provan discusses the impact of internal organizational power on strategy decisions. Due to the far reaching consequences of technological change, it is reasonable to consider information systems development as a strategy decision. He defines power as the ability of a department to influence the formulation and implementation of strategic-level decisions in a way that is supportive of its needs and perspective.[13] By adapting this definition, it becomes obvious that a department in good standing within the organization could have ultimate control over IS development decisions.

The strategic decision process is influenced by major groups in the organization with the most power. This suggests that the impact of external environment factors may be diminished, and that strategy decisions are based on the internal factors at work in an organization.

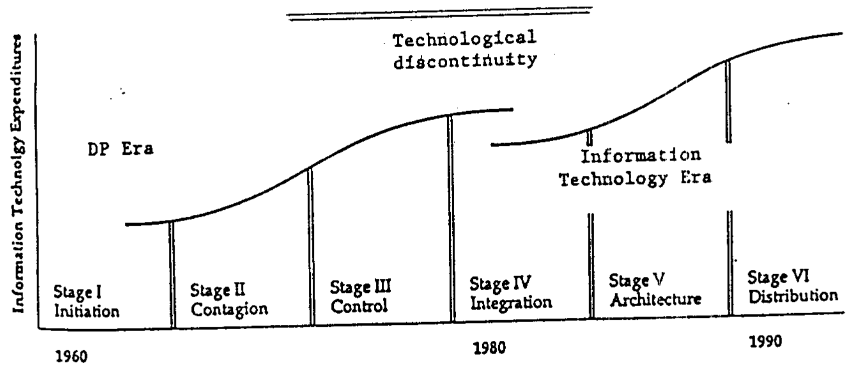


Figure 3: Nolan's Stage Analysis

However, power and tradition may play a more direct role in the allocation of resources and in the formulation of strategic decisions. The IS manager must be aware of both the internal and external issues and attempt to reduce the impact on the IS organization. Consideration must be based on the organization as a whole, not just suboptimization of the best department.

Many problems arise due to the fact that IS has not been in the organization as long as other departments. It does not have the tradition behind it. This internal focus could cause an organization to overlook certain technological advances that might enhance their competitive position.

Another factor to consider is the organizational position of IS. Depending on the size and structure of the organization, the IS position can be anything from a support department to the main thrust of the business. In fact, sometimes it may not even be a department as such. This position can determine the extent to which information resources are utilized. If IS is considered a support department, the typical areas will be automated: accounting, payroll, purchasing, etc. These areas require simple data manipulation which requires primarily technical skill in order to convert existing practices into automated ones. If, however, IS is in a strategic position the organization, automated decision making, or Decision Support Systems, usually results. This requires systems professionals to be versed in business processes and have good communication skills in order to automate these major management cognitive processes.

The position of Information Systems changes in an organization as time passes. As technology is integrated into the organizational framework, the emergence of IS assumes a more strategic role. As the role of IS changes, so will the skills required by the IS professionals involved in the operation. This will not happen, however, unless organizations change internally and accept IS as an important link in strategy and decision making.

ENSUING PROBLEMS

The foregoing events have surfaced many significant problem areas which need to be solved and managed. To bring this into focus, it is well to track the turning point from the DP Era to the Information Technology Era as shown in Nolan's stage analysis. In the author's view, Nolan accurately projected this transition as shown in Figure 3.[14]

As he did in his model, we should now be concentrating on managing the advanced stages of computer technology which are in the IS era. He points out that technological discontinuity occurs near the end of Stage III Control which is the critical point where preoccupation with the computer must give way to what the system can provide to the enterprise through integration, architecture, and distribution of the output. In this case, this means choosing enterprise over entity, or the "big picture" over the technical details in the focus of management's requirement for information for the organization's needs and purposes.

<u>Issues of Change and Conflict</u>--There are human elements inherent in an organization. Their impact on IS development can be categorized in many ways. The fast changing nature of technology requires looking into human and organizational reaction to change and conflict. For example, an employee who continues to manually post accounts after an automated system has been introduced into the organization. Resistance can be minimized if an organization preparing to undergo change keeps employees informed along the way. If the users are involved all the way, uncertainty will be decreased and the employees will also give input as to the best disruptive way to institute the change. This

emphasizes the need for the IS manager to be aware of personal issues involved in planning and changing information systems.

What does this mean for IS? Essentially, it implies that information systems departments need to establish themselves in the organization in such a way that they support the common goals of the organization. Both issues, change and conflict, can be supported by the CSF method mentioned earlier provided that the appropriate structure exists. Through identifying the critical success factors, organizations can move to the future.

Problems in Assimilating Technology--Technological assimilation into the organization is difficult. One reason is that IS technology has had a short life. It cannot be as well established in an organization as the financial functions are because it has not been around as long. Another reason may be that the field has undergone dramatic evolution in its technologies. What used to take years to go from research to use now can take only minutes. With every new system installation, there are newer and better systems being designed. With the current exploration of expert systems, the potential becomes almost unlimited.

Another reason may be the complexities in IS development which force creation of specialized departments and strain relationships. These specialized departments require increased specialization and generate a greater need for skills from IS professionals.

Organizations are not structured to change as quickly as technology does. This has strategic implications for the organization of the future as it attempts to assimilate technology into its core. Conversely, organizations that do not change structurally may not be able to take full advantage of the technology.

There is a solution to many of the problems addressed here. This is to take a top-down approach and have an Executive Information System (EIS). EIS and MIS, when used together, can be powerful tools. MIS have integrated the functions of data reporting, data storage and retrieval, and decision support. EIS have the potential of supporting executive managers by pointing the way to problems in need of solutions.

PROLIFERATION OF INFORMATION MANAGEMENT

Without fully realizing the rapid pace of change, professionals are faced not only with developing the connectivity between mainframe and PC systems, but are now having to deal with another area of data communications sometimes referred to as Networking. In a recent 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."[15] No longer is communications a separate field from data processing; it now is a part of distributed processing. Networks began to matter when they changed the behavior of the attitudes and dialogues among managers. Data processing personnel and MIS managers found that Local Area Networks (LANS), Wide Area Networks (WANS), satellites, special hardware and software technology, and protocols are now needed for integration with the MIS. This calls for close coordination and a requirement for new architectures. This is basically where we are today.

CONCLUSIONS AND IMPLICATIONS FOR THE FUTURE

- The rapid rate of emerging technologies is making the survival role of Information Systems more and more difficult.
- A business approach and a timely huge leap in testing, accepting and adapting new technologies is needed for a competitive edge in IS.
- Managing change is the order of the day (or more accurately the decade).
- Adapt database and access languages that are powerful and flexible so that new applications can be quickly developed.
- IS managers should never lose track of the enterprise perspective and be alert to the constantly changing needs of the *Shareholders*.
- Be mindful of new developments such as the *Data Highway (or Information Super Highway)* so that more effective strategy can be planned for the future.
- The primary role of IS, although being challenged, is broadening and is moving from technician to manager/consultant/educator.
- The quality and focus of IS education both in schools and in the work place will play a prominent role our future. We must keep our priorities straight and always look ahead--not back!!!

References

1. Steven Alter, "Where is our Future in Business Schools: Information Technology or Business Processes?" *Proceedings of the Tenth Information Systems Education Conference*, sponsored by DPMA Education Foundation, Phoenix, AZ, November 1993, p. 127.

2. *CIO, Magazine for Information Executives*, February 15, 1994, Cover.

3. Eric Byers, presented "Client/Server Implementation Plan," *Digital Equipment Corporation*, January 1994.

4. James Martin, *Fourth-Generation Languages, Volume 1 Principles*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1985, p. xi.

5. James Martin, *An End-User's Guide to Data Base,* Englewood Cliffs, NJ: Prentice-Hall Inc., 1981, p. 75.

6. Andy Reinhardt, "Building the Data Highway." *BYTE*, March 1994, pp. 46-74.

7. *COMMUNICATIONS of the ACM,* (Artificial Intelligence Issue) March 1994--Volume 37, Number 3.

8. C. Bruce Kaven and Houston H. Carr, "An Overview of Electronic Data Interchange," *Handbook of IS Management*, 1992-93 Yearbook, Robert E. Umbaugh, Ed, Boston, MA: Auerbach Publications, pp. S-81-S-91.

9. M. Lynne Marcus, "Power, Politics, and MIS Implmentation," *Corporate Information Systems Management:: The Issues Facing Senior Executives*, Homewood, IL: Richard D. Irwin, Inc. 1983.

10. John F. Rockart, "Chief Executives Define Their Own Data Needs," *Harvard Business Review, (March/April 1979).*

11. J. F. Rockart and Adam D. Crescenzi, "Engaging Top Management in Information Technology," *Sloan Management Review* (Summer 1984, pp. 3-16.

12. Kevin Crowston, Thomas W. Malone and Felix Lin, "Cognitive Science and Organizational Design: A Case Study of Computer Conferencing," *Human-Computer Interaction 3* (1987-1988), pp. 59-85.

13. Keith G. Provan, "Environment, Department Power, and Strategic Decision Making in Organizations: A Proposed Integration," *Journal of Management,* 15 No. 1, (1989), pp. 21-34.

14. Richard L. Nolan, "Managing the Advanced Stages of Computer Technology: Key Research Issues," in F. Warren McFarlan, ed., *The Information Systems Research Challenge,* Boston, MA: Harvard Business School Press.

15. Ram Charan, "How Networks Reshape Organizations--For Results," *Harvard Business Review* (Sep/Oct 1991), pp. 104-115.

PERSONALITY TYPE ISSUES IN IS EDUCATION

Eugene Kaluzniacky
Business Computing Program
University of Winnipeg

Catherine Bishop-Clark
Information Systems
Miami University, Ohio

ABSTRACT

It is currently an accepted principle that MIS is a socio-technical field in which personality characteristics of the people involved in the area do play an influential role. Since different personality types involved in IS work tend to perceive their work and their roles differently, it is a noteworthy issue for IS educators to examine how IS students with different personality types perceive and perform in different IS courses taught by different approaches. This workshop will introduce the participants to the Myers-Briggs personality type indicator. It will then expose them briefly to published research highlights regarding the applications and benefits of this indicator in IS work.

As the participants begin to appreciate constructive insights which the MBTI can provide, the presenters will each highlight results from their research on MBTI application in teaching of several IS courses in their respective institutions. The workshop will then take on a participative nature, as attendees will receive exercises in the teaching of specific IS topics to different personality categories. There will be considerable opportunity for discussion. Finally, the benefits of making senior IS students specifically aware of their personality type and the possible impact of this awareness on their careers will be examined.

A Systems Approach to a Data Structures Course for Information Systems Students Consistent With DPMA IS'90

Michael V. Doran
Herbert E. Longenecker, Jr.
J. Harold Pardue

University of South Alabama
School of Computer and Information Sciences
Mobile, Alabama 36688
(205) 460 - 6390
doran@cis.usouthal.edu
bart@cis.usouthal.edu
hpardue@garnet.acns.fsu.edu

ABSTRACT

The DPMA [DPMA 1981; Longenecker and Feinstein 1991] and the ACM [ACM 1981] have defined and updated [IS'90: Longenecker and Feinstein 1991; IS'94: Longenecker, Reaugh, et al 1994] undergraduate Information Systems (IS) curriculum models. IS'90 and IS'94 provides a well structured systematic process to accomplish the overall IS curriculum goals and objectives. In these documents, course sequences are identified and specific target objectives with associated knowledge elements are enumerated to specify desired levels of instruction. This paper reports on a specific implementation of a data and file structures course sequence using the DPMA IS'90 methodology. The course contents is based largely on the IS2 course of the ACM'81 model which embraces many of the knowledge elements associated with IS'90 cluster D objectives. The course we describe, Data Structures for Information Systems (DS-IS), is the capstone part of a sequence of courses which serves as a foundation in our IS graduate program. Initial prerequisite courses to our graduate DS-IS course are the undergraduate CS1 and 2 [ACM 1991] sequence. The entire sequence, including the graduate DS-IS course, is viewed and structured as a system. This paper reports observations of student performance in this system of courses.

INTRODUCTION

The DPMA IS'90 model curriculum document provides guidelines for the structuring of degree programs as well as courses. The methodology identifies broad goals, breaking the curriculum top-down into segments called "clusters" and then associating cluster target objectives with knowledge elements. These clusters define a sequence, or architecture which is <u>an</u> implementation of the overall curriculum goals. The clusters are described by goal statements which state the plan for the cluster. These goals are then actualized by introduction of target objectives which describe and enumerate exit behavior of the cluster. These target objectives detail specific associations of knowledge elements which must be learned to a given level of depth. A cluster can be factored into one or more courses which are also broken into sub-clusters each with goals, target objectives with associated knowledge. We refer to this process as a cognitive modeling technique.

IS'90 and IS'94 both cover the same general knowledge domain. In fact, a graduate of the IS'94 model curriculum should be able to articulate into the E, F and G clusters specified in IS'90. In IS'94, specific courses are described very differently from those of IS'90. However, both sequences actually covering clusters, A through D achieve a similar subset level of knowledge. The similarities of exit knowledge levels at the D Cluster level are the basis for the claimed feasibility for articulation between two and four year programs.

MODEL COURSE SEQUENCE

This paper explores the use of the traditional computer science [ACM 1991] one year sequence of CS1 and 2 [Koffman 1984; Koffman 1985], coupled to an implementation of the ACM 1981 IS2 data structures course which includes much of "Cluster D" of DPMA IS'90 and IS'94.

At our university, most of our Information Systems graduate students

Graduate Degree Program in Information Systems

Prerequisites	"CIS CORE"	IS Graduate Program
CIS 140,141,142 --> (ACM-CS1,2)	Data & File --> Structures	"Comprehensive Exam"
	Operating Systems	Information Analysis Systems Design Systems Development IS Audit Controls
	Database	IS Policy IS Management Telecommunications
	IS Principles	Decision Support Systems

Figure 1. Degree Program in Information Systems. A CS-1,2 sequence is a prerequisite to the CIS Core, MS Comprehensive and IS Courses.

seeking the masters degree have undergraduate degrees <u>not</u> in the computing sciences. An important first step for most of these students is the development of computing foundation skills in four core areas including (i) operating systems, (ii) data and file structures, (iii) database and (iv) principles of information systems (see Figure 1). While students who have an undergraduate CIS background can easily and immediately enter the graduate core sequence, the majority of our students require prerequisite skills which are developed in our programming courses. At the completion of the graduate core sequence, students must take and pass a written comprehensive exam covering these core course topics. Upon successful completion of the exam, students take the remaining graduate courses to fulfill the degree.

It had been observed that problems existed in the process. Failure rates on the comprehensive exams were becoming high and unacceptable; students took core courses without completing the prerequisite sequence or students took additional graduate courses without first completing the core courses. It was identified that much of the problem centered on the data and file structures for IS (DS-IS) core course of the graduate sequence and the undergraduate courses serving as prerequisite to it. Therefore, we chose to study this sequence as the model to apply the concepts of system theory and educational metrics as identified in the DPMA IS'90 document [Longenecker and Feinstein 1991].

The system identified should take as input a BS undergraduate student and produce a graduate student capable of passing the comprehensive exam in the area of data and file structures. The sub-systems consisted of four courses: (i) Computing I and II - CIS 140, 141, and CIS 142 and (ii) Data and File Structures - CIS 508. The three 100 level courses form a year long sequence (at our university a quarter is 40 lecture hours) which covers the goals and objectives of the application development cluster of the DPMA '90 document. It likewise satisfies the Computing I and II goals and objectives of the ACM'91 document. CIS 508 is the IS graduate data file structure course (DS-IS).

The purpose of the exploration is to *apply* the IS'90 cognitive modeling techniques to facilitate *an* implementation of the curriculum objectives and to show that the expansion is consistent with the DPMA IS'90 methodology and with many of the curricular objectives. Also, we believe that the DPMA curriculum methodology can be used to establish a basis for educational outcome assessment measures. We propose that these measures can be based upon the cognitive framework contained in the expressed target objective and knowledge element behaviors. We also believe and hypothesize that the concepts of general system theory can be employed to ensure a strategic

DPMA Knowledge Level		Discussion of Level and Concepts of Assessment
1	Recognition	Recognition knowledge is the ability to identify facts. It is accomplished by learners through reading, lectures and discussion. Students should be able to recite definitions and facts. Exercises, homework and quizzes enable students to practice and illustrate this knowledge. "Fill-in-the-blank" format exams are appropriate.
2	Differential	Differential knowledge implies having mastered the ability not only to recognize, but to identify facts and their relationship in a network of facts. Identification of facts within a context is an expectation of students at this level. Lecture and lab/discussion, or computer based training formats will enable learners to master this level. Matching, fill in the blank, or sentences which illustrate comparison are appropriate methods of assessment.
3	Comprehensive	Comprehensive knowledge implies that students can USE their knowledge. "Telling a story", or other forms of explaining are required, as is the ability to apply with direction the knowledge. Levels 1,2 are clear prerequisites. While lecture may be used to give examples, more concrete forms of problem solving in laboratory are needed. Using a program language or design method are examples. Evaluation is accomplished by "giving a requirement, then produce ..." format of questioning.
4	Application	Application knowledge implies the ability to search a variety of KNOWN methods and to select and use the appropriate method to solve a problem. The learner should be given no explicit cue to aid in determining the method of search, selection and use of given knowledge. Considerable experience at level 3 is prerequisite, as is much exposure to repeated exposure to appropriately focused problem solving environments. Students must be instructed in the search strategies, and must have mastered the comprehensive level through many successful trials. This skill level is sometimes only mastered with "on-the-job-training". Assessment of this level of skill requires watching the student work in a problem solving situation and observing the appropriate searching and selecting behaviors in the context of real problem solving.
>5	Analysis, Synthesis, Evaluation...levels not expected in our class environment	

Figure 2. Definition of Knowledge Levels and Assessment Concepts.

sequence of learning modules which define the DS-IS course sequence.

The degree to which behavioral objectives have been reached by the student learner can be verified both through self or faculty examination of student behaviors. Then, by comparing results of such quality verifications, continuous feedback and progress evaluation can be provided to the learner. Self examination of performance with respect to clearly communicated goals and objectives can be accomplished by the student. This verification requires precise up-front communication to the students of desired knowledge, knowledge measurement methodology, and a learning time frame. The resultant feedback provides the learner timely and precisely identified rewards as well as suggestions for improvement. Such continuous feedback is empowering, and establishes confidence in the student's perception of accomplishment and ensures mastery of the target objectives and associated knowledge elements.

Therefore, this study augments the DPMA curriculum guidelines by implementing the DS-IS sequence of behavioral objectives and demonstrates techniques usable for monitoring and evaluation of outcome performance.

COURSE SEQUENCES AS SYSTEMS

We take the view that a sequence of courses can be considered as a system. The concept of implementation of such a system and has been applied to educational systems [Gagne 1985; 1988; Longenecker and Feinstein 1991; Longenecker and Reaugh et al 1994]. Because ideas can be introduced in one component of a system and followed up in a later one, the concept of "spiraling" [Argyris 1976; 1977; Feinstein and Longenecker 1991; Longenecker and Reaugh et al 1994] has been used to describe revisiting ideas. In order to characterize the system output, we have utilized the cognitive scale of Bloom [Bloom 1950; 1956] as described in the DPMA IS'90 curriculum document [Longenecker and Feinstein 1991] (see Figure 2). By setting specific objectives, which have attached cognitive metrics, the effectiveness of the system can thus be measured by determining if student behavior meets or exceeds the metric. To the extent that students achieve the desired objective, the system is said to be effective. With respect to a given system, a subsequent coupled system is called a "client" of the preceding system. The client system expects and indeed requires that its supplying system produces precisely specified output meeting its well defined needs/objectives. The client system makes this demand since it uses the output of the supplying system to

Data and File Structures for IS (DS-IS) Objectives

Class	Topic and Entry/Exit Levels of Coverage within Class
1	problem solving 4/4, algorithm design 4/4, control structures 4/4, arrays 4/4, records 4/4
2	abstract data types(ADT)0/1
3	ADT 1/2, list 0/1, stack 0/1, queue 0/1
4	ADT 2/3, list 1/2, stack 1/2, queue 1/2
5	ADT 3/3, list 2/3, stack 2/3, queue 2/3, pointer 0/1
6	string 2/3, file 2/3
7	string 3/3, file 3/3, recursion 2/3
8	sorting 1/2, exchange sort 1/2, selection sort 1/2, insertion sort 1/2
9	sorting 2/3, exchange sort 2/3, selection sort 2/3, insertion sort 2/3, quick sort 1/2
10	searching 1/2, linear search 1/2, binary search 1/2, hashing 0/1
11	searching 2/3, linear search 2/3, binary search 2/3, hashing 1/3
12	pointers 1/3
13	trees 1/2, pointers 3/3, arrays 4/4, list 3/3
14	trees 2/3, recursion 3/3
15	trees 3/3, files 3/3, problem solving 4/4
16	problem solving using pointers 3/4, multilinked lists/trees 3/4
17	problem solving using file structures 3/4

Figure 3. Course Topics and Target Objectives for the Data and File Structures for Information Systems Course.

ensure its own output. Without this demand, the client systems output will be flawed. Thus, the exit objectives for a given component are at least the equivalent of the prerequisites for the next component in a coupled system. We will introduce some data later describing the results of meeting or failing to meet such prerequisites requirements.

SYSTEMS THEORY AND LEARNING CONCEPTS

General system theory defines a system as a collection of purposeful interrelated components which work together to achieve a goal [Churchman 1968; Atihuv 1989; Richey 1986]. A clear and explicit set of system goals are required for a desired transformation of an input to an output. The outcome of each subsystem can serve as input to coupled systems. We identify a component of a system as a mechanism for causing learning to occur. Students possessing entry levels of knowledge serve as input to the system component. Following the "process of learning", system output is equivalent to the student possessing demonstrable and precisely specified exit knowledge level. Learning can be achieved within the system through a well planned sequence of subsystems [Gagne 1985; 1988; Salisbury 1989]. Each subsystem attains an incremental goal. Progress through additional components ultimately leads to the overall goal, thereby establishing the overall effectiveness. In addition, an ideal system must be meeting the stated goals and objectives by efficiently using the available resources including texts, notes, fellow student time, instructor time, lab time, and perhaps other factors.

EDUCATION CONCEPTS AND METRICS

Educational concepts provide the basis for the assessment of outcome performance. The basic mechanics of the outcome assessment measure can utilize conventional measures of student performance, e.g., programming assignments, tests, and quizzes. What distinguishes this approach from previous work is the explicit use of precisely stated behavioral objectives which are explicitly examined by the conventional measures. This ensures conformance to the curriculum specification. This mapping is the key factor in providing a quantifiable measure of a student's incremental progression through the curriculum.

Long established metrics for the assessment of student performance exist in the discipline of education (see Figure 2). The metric used in the model described in this paper was the Taxonomy of Educational Objectives [Bloom 1956] as well as in IS'90 [Longenecker and Feinstein 1991]. As an outcome assessment measure, Bloom's approach has been shown to be highly successful [Chance 1987]. In the taxonomy, learning objective achieved and goals are clearly defined in terms of six levels of knowledge. Each level consistently and accurately measures the degree or quality of a student's proficiency with a given body of knowledge. Quality is defined as the degree to which outcome performance consistently matches or exceeds expectations.

A significant contribution of Bloom's Taxonomy of Educational Objectives to the current work is a quantifiable, standardized framework for evaluating and monitoring students' progression through the curriculum.

A taxonomy is a classification of entities according to their natural relationships. Bloom's six level taxonomy defines the relationship between learners and an increasing set of levels of cognitive tasks. Learners exhibit level one behavior which act as a prerequisite to level two behavior; level two task behavior is likewise prerequisite to level 3 and so on. For each knowledge element, there will be explicit target levels and specific student behaviors which represent the intended outcomes of the educational process.

Each level defines characteristics of behavior which the learner is expected to demonstrate. At level 1 (Knowledge), the student should know and can recite, recognize (DPMA level 1) and differentiate facts (DPMA level 2) on a given topic. To demonstrate level 2 (Comprehension), students should be able to paraphrase, translate, interpret, extrapolate and otherwise **USE** the facts about a topic when given necessary cues. Application is demonstrated at level 3 whenever students can apply the facts to solve a problem in a new situation without cues. Level 4 (Analysis) occurs when students can define the relevant components of new abstractions. Level 5 (Synthesis) is the organization, development and appropriate usage of new abstractions by the students. Finally, level 6 (Evaluation) is mastered when students can evaluate the effectiveness and efficiency of alternate syntheses. The target levels considered for the courses in the sequence were level 1, 2 and 3.

Bloom's Taxonomy of Educational Objectives provides the basis for a quantifiable framework for sequencing the presentation of the curriculum knowledge units. However, the actual learning sequence is not considered to proceed in discrete stages but rather in a continuous upward spiral. This spiralling progression represents a review and reinforcement of previously presented materials via the introduction of new concepts.

COURSE DEVELOPMENT

Initial work involved the specific mapping of topics which were covered in the course to a specific Bloom level of knowledge. It was generally recognized that for most topics the target Bloom level would be 2 or 3. Initial efforts to attempt above level 3 and sustain that level were determined to be unrealistic. With this target defined, the topics were sequenced throughout the courses with incremental increasing levels specified (see Figure 3). This provided a mechanism to define exit objectives for one course, which thereby served as entry behavior for the following course. This simple fact from system theory, that the result of one sub-system acts as input to subsequent sub-systems, gives student and faculty an explicit metric to assure that topics targeted for the current course can be utilized with the expected prerequisite knowledge. This fact also provides a framework, via a placement exam, to evaluate the readiness of a student who wishes to bypass certain courses.

Next, exercises with the specific target objectives for each topic were defined. Again, in keeping with system theory, constant feedback and monitoring of the system is necessary. This is especially true when viewing educational courses as systems. For example, consider presentation of a WHILE loop: In CIS 140, the WHILE loop was covered to level 2 (comprehension). Learning objectives therefore would include the ability of the student to use a WHILE loop given cues, as well as paraphrase, translate, interpret the use of the construct. It is also inherent that level 1 knowledge be present, which includes factual definition. Appropriate lecture and readings were scheduled to first achieve level 1. Homework and quizzes were performed to assure that level 1 was achieved before level 2 was attempted. This is in accord with the general tenets of system theory where achievement of this individual topics might itself be viewed as a sub-system. After mastery of level 1 facts regarding a WHILE loop lectures were delivered and problems were assigned which would facilitate mastery of level 2. Feedback was provided via homeworks, quizzes and programming assignments. By continuous monitoring of performance, adjustments were made. Progress was not halted for one or two

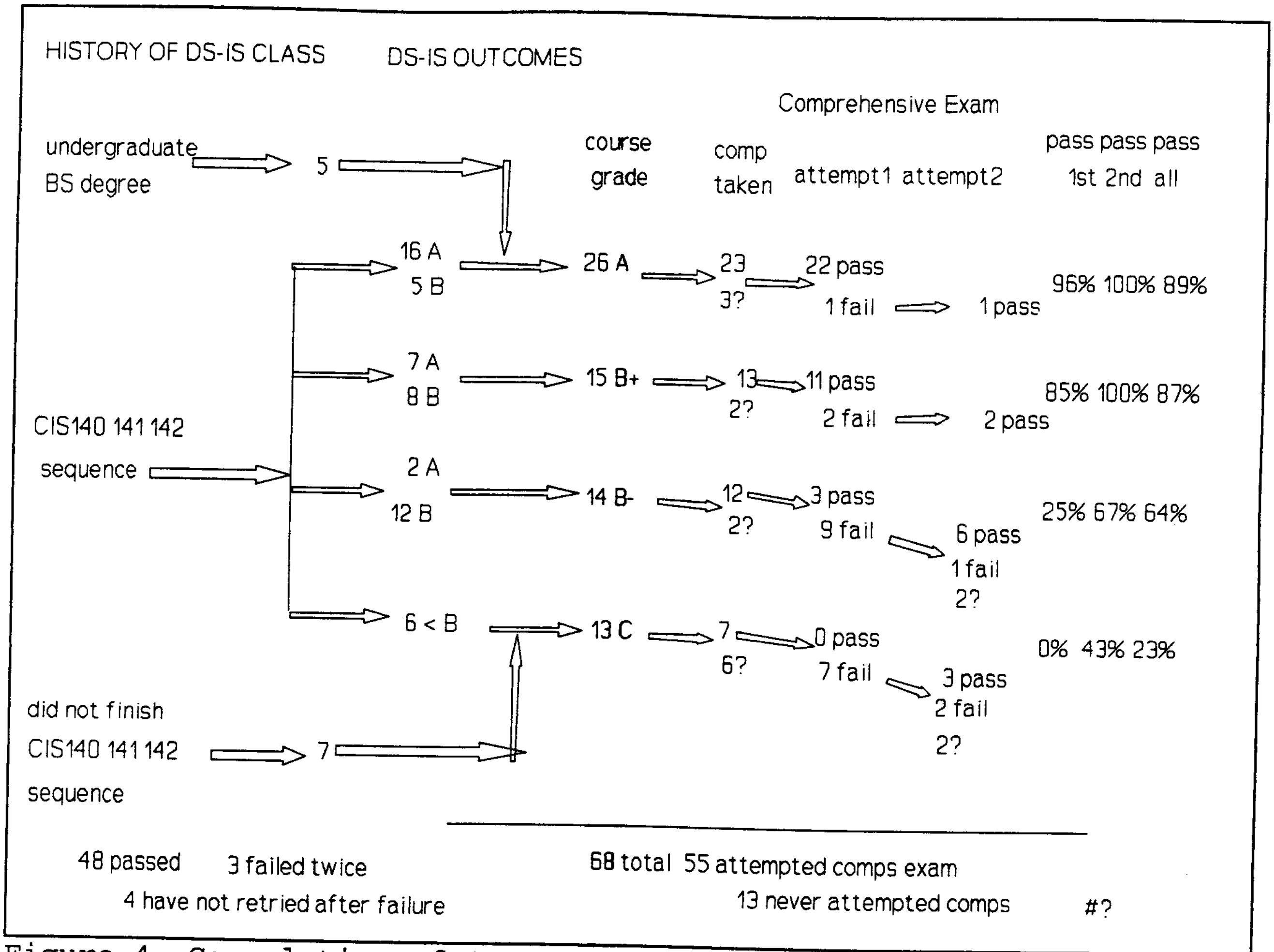


Figure 4. Correlation of Course Grades with Comp Exam Results

students to catch up; in such cases individual remediation is required and encouraged. However, if for any reason mastery is not achieved by a significant portion of the class corrective measures was taken, and appropriate topics were recovered. In all cases, students were made aware that each topic is necessary for overall success, and that any topic not mastered to the target level will lead to problems in the future because of the knowledge deficit.

It was also recognized that due to the spiral nature of learning that topics would be revisited. This was accounted for not only with the assumption that exit mastery level would be maintained but in some cases extended, due to this spiral revisiting designed to achieve higher levels. This was the case with several topics in DS-IS (CIS 508) where level 4 was expected. This major undertaking was performed for each topic in the 4 course sequence, with special emphasis on those topics of the CIS 508 course. An added feature included in the CIS 508 course was an explicit explanation of the process and a schedule of topics with target levels being included in the syllabus (see Figure 3). This explicit "roadmap" was thought to be of more benefit to the more mature graduate student and also serve as an explicit guide of expectations for the comprehensive exam.

OBSERVATIONS

The definition of the courses with the principles of system theory and Bloom levels as the driving factors allowed for clear and unambiguous expectations of student behavior. Therefore, the expected entry behaviors of students and likewise the exit proficiency students must demonstrate, was clearly communicated to all. With regard to the comprehensive exam, an absolute level of mastery was defined with clear and certain objectives.

In the CIS 140, 141 and 142 courses, it was observed that with the instructor constant and students continuous mastering the topics, a proactive constructive learning environment was fostered. Additional studies with the results published elsewhere indicate a high level of satisfaction by the students and a statistical improvement in their test scores.

The major observation comes from the CIS 508 course. In previous versions, a high percentage of students often failed the comprehensive exam. With the new model in place, several improvements occurred.

(1) First, an entry exam was administered in the first class with the entry behavior clearly defined. Those students not meeting the expectations were advised what measures would remedy the deficit. Any student choosing to ignore the advice proceeded at their own risk. It was often the case that students who failed to adhere to the foundation prerequisite structure were those with the deficit. Those who had not successfully completed the foundation sequence were unaware that the exit behavior mastered previously was the key to success in the current sub-system. Those who were successful were in the foundation sequence were equally made aware of the topical continuity.

(2) Milestones could be identified as to progress to successful completion of the course and ultimately the comprehensive exam. This gave the students explicit targets to aim for and a signal of when those targets were achieved.

(3) There was a very direct correlation between performance in the CIS 508 class and results of the comprehensive exam (see Figure 4). Almost all those students achieving a grade of "A" in the course passed the exam the first time (in the past four years only one "A" has had to take the exam a second time). Most students (approximately 85%) who earned a strong "B" (above 85%) passed the first time. Those students earning a weaker "B" experienced more problems passing the first time, usually half passed the first time. Students earning a "C" (only one "C" is allowed as passing in graduate school), failed at the rate of 100% on the first test and often the second. It was often the case that these "C" students were also those identified as having a deficit on the entry exam and either chose to ignore advice or still failed to achieve the necessary target levels. These observations serve as fair warning to the students of what the expectations are and the standards expected on the comprehensive exam. However, that doesn't stop many individuals from continuing to ignore the warning and attempt the course with the stated deficit. During the last session of the course, 6 students were judged as having weak or missing prerequisite knowledge. They continued in the course and earned a "C". Of those students, 3 have taken the comprehensive exam and failed the first time. Of those 3, two have retaken and one failed again, the other passed after much individual effort and retaking additional courses.

CONCLUSION

It was hypothesized that the curriculum could be implemented using the principles of system theory and educational objectives to insure a stable learning environment. One sequence was identified at our university and defined in such a fashion. Each course was treated as a sub-system which contributed to the overall exit behavior. This sequence of interrelated subsystems was communicated to the students. The feedback observed and monitored was measured in terms of explicit Bloom levels of knowledge. These levels were likewise made known to the students so that behavior might be recognized when successfully completed. Finally, a course sequence was implemented that could reasonably predict success on the exit comprehensive exam. Students could track their progress in light of the target exit objectives. This did not completely eliminate all problems, however, there is now an audit trail of where students have failed to achieve a target and the burden is clearly on them to take corrective action, if they so choose.

In summary, just as it is a central goal of the overall curriculum to develop management and organization skills which should be applied to software systems, these same skills can and should be applied to the educational process. Our conclusion is that education is a system which requires monitoring and control at each stage of development. A clear and explicit set of standards must be in place to measure the effectiveness and efficiency of the process. Bloom levels provide a set of standards to measure mastery; system theory provides a framework for the application and organization of the process.

BIBLIOGRAPHY

[ACM 1981] ACM Educational Programs and Information Systems.

[ACM 1991] ACM/IEEE Joint Task Force Computing Curricula 1991.

[Argyris 1976] Argyris, C., "Single Loop and Double Loop Models in Research Design on Decision Making", Administrative Science Quarterly, Vol. 21, No. 3, September 1976, pp. 363 - 375.

[Argyris 1977] Argyris, C., "Double Loop Learning in Organizations", Harvard Business Review, Vol. 55, No. 5, Sept/Oct. 1977, pp. 115 - 125.

[Atihuv 1989] Atihuv, N. and S. Neumann, Principles of Information Systems for Management, W. C. Brown, Dubuque, Iowa 1989.

[Bloom 1950] Bloom, B. S., L. J. Broder, Problem-solving Processes of College Students: An Exploratory Investigation, The University of Chicago Press, Chicago, Illinois, 1950.

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

[Chance 1987] Chance, P., "Master of Mastery", Psychology Today, April 1987, pp 43-46.

[Churchman 1968] Churchman, C. W., Systems, Dell Press, New York 1968.

[DPMA 1981] The 1981 DPMA Curriculum for Undergraduate Information Systems Education.

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

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

[Koffman 1984] Koffman, E. P., P. L. Miller and C. E. Wardle, "Recommended Curriculum for CS1: 1984 a Report of the ACM Curriculum Task Force for CS1", Communications of the ACM, Vol. 27, No. 10, October 1984, pp. 998-1001.

[Koffman 1985] Koffman, E. P., D. Stemple and C. E. Wardle, "Recommended Curriculum for CS2: 1984 a Report of the ACM Curriculum Task Force for CS2", Communications of the ACM, Vol. 28, No. 8, August 1985, pp. 815-818.

[Longenecker and Feinstein 1991] Longenecker, H. E. and D. L. Feinstein, Information Systems: The DPMA Model Curriculum for a Four Year Undergraduate Degree, IS'90, DPMA, Park Ridge, Ill.

[Longenecker, Reaugh et al 1994] Longenecker, H. E., W. R. Reaugh, R. Fournier and D. L. Feinstein, Information Systems: The DPMA Two Year Model Curriculum For IS Professions, IS'94, DPMA, Park Ridge, Ill.

[Richey 1986] Richey, R., The Theoretical and Conceptual Bases of Instructional Design, Nichols Publishing, New York 1986.

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

DESIGNING TECHNICAL MBA/MIS PROGRAMS

Alden C. Lorents and Greg Neal
College of Business 15066
Northern Arizona University
Flagstaff AZ 86011
602-523-3510 602-523-7364

ABSTRACT

Technical MBA programs are emerging to meet the demands of industry. Many MBA programs are currently under redesign because the traditional general MBA had dropped in popularity with corporate recruiters. However, schools with MBA degrees that have a technical emphasis such as computer related courses are experiencing excellent results with the placement of their graduates. We are currently redesigning our MBA program with the intent that the primary output of the program would have an emphasis in a technical track. This paper looks at the preliminary design of an MBA program with two MIS technical track options.

INTRODUCTION

Industry demand for the general MBA has slipped over the past few years while the demand for MBA graduates with technical skills continues to rise. Business schools across the nation are busy revising their MBA curriculum to make their graduates more competitive for the job market. These revisions are going a number of different routes that include varying program length, the integration of course content from multiple disciplines, the use of internships and company sponsored projects and a focus on topics of high interest by corporate recruiters. Many schools are experiencing multiple job offers for graduates with a fair amount of computer background, while students in the same program without the computer background are getting very few job offers and are sometimes without a job for up to a year. MBA/MIS programs have been around for a number of years. However, most MBA/MIS programs tend to be more management and decision support oriented. Very few MBA/MIS programs tend to be technically oriented.

A recent Business Week article talks about the new emerging MBA referred to as "Techno-MBAs" [1]. These are MBAs with engineering and/or computer related disciplines. "Right now , they number only about 3000 of the 75,000 MBAs that graduate each year." A recent issue of Money magazine puts systems analysts at the top of the list for jobs with the best potential for next five to ten years. "The Bureau of Labor Statistics (BLS) believes there will be 501,000 systems analysts jobs created between now and the year 2005, a gain of 110% from today's 455,000--and that 501,000 job forecast represents a 37% upward revision from just two years ago." [7]

REENGINEERING THE MBA PROGRAM

This purpose of this paper is to look at the potential design of an MBA program with multiple technical MIS tracks. The redesign of the program would basically eliminate the general MBA. The reasons for this move is to develop a niche MBA program that will develop a good reputation measured by recognition and a high percentage of quality job placements upon graduation.

CURRENT MBA PROGRAM

The current MBA program has been in place for a number of years with two tracks, the general MBA and the MBA/MIS. The program is a traditional one year program assuming the student has an undergraduate business degree. The program requires the prerequisites, Introduction to CIS taught at the 100 level and the Introduction to MIS taught at the 300 level. In addition the MBA/MIS emphasis students come in with an Introduction to COBOL. The current program requires 33 semester hours that covers a 5-week summer session, two semesters and a 3-week pre-summer session. The MBA/MIS emphasis students take twelve of the 33 hours in technical information systems courses. These students have typically entered the job market in system development positions with companies like Andersen Consulting, American Management Systems, Hewlett Packard, American Express, Sandia Labs, and Shell.

REDESIGN OF THE MBA PROGRAM STRUCTURE AND CORE

Like other MBA programs across the country, the current MBA program is currently under review to redesign its format and content. The current program planning is operating under the following constraints that have been set by the program redesign committee. [3]

1. The MBA program will continue to be a one year program for students who have an undergraduate degree in business and have the prerequisites for the program.

2. Students will enter the program generally in the first 5-week summer session.

3. The program will allow part time students, but they must commit to at least half-time blocks and complete all summer session content before participating in the semester blocks.

A summary of the program structure is shown in the table on the next page. The content of the program during the summer will concentrate on leveling, tools (statistics, economics, accounting and finance), communications and presentations, case training, group dynamics training, and computer training (Hardware, DOS, Windows, Microsoft Office, Networks, and Internet). The format for the Fall and Spring semesters will be set up as parallel blocks; a core block and an emphasis block. Full time students will enroll in the core block and select one emphasis block. A part time student will normally take the core block one year and select an emphasis block the next year. The time allocated between core and emphasis blocks will be split evenly over the two semesters.. There will also be a 3-week core component after Spring semester and before the summer sessions that will concentrate on topics related to the future. This will include issues that MBA graduates may be confronted with in the first five years after they graduate. Topics may include global economic issues, gender and family issues, electronic commuting, and emerging trends. The core will be heavily integrated and driven primarily by cases and projects. The content of the core will concentrate on two general areas: decision making models and tools, and human resource dimensions. These two areas will include topics such as developing business plans, strategic models, time based competition, global markets, TQM, business process reengineering, information system planning, ethics, environmental issues, training, leadership, followership, empowerment, and negotiation. Many of the cases used will involve technology related products and organizations. Most cases will integrate three or more traditional disciplines within the College of Business. Skills and tools needed to work on the cases will be emphasized in the leveling and tools related courses in the summer.

MBA/MIS TECHNICAL TRACKS

MIS TRACK ONE - ENTERPRISE WIDE INFORMATION ENGINEERING

Two MBA/MIS technical tracks are currently in the development and planning stage. The first track would evolve from the current MBA/MIS program. This track will be a system development track with an information engineering emphasis using IEF. Students graduating from this track would normally go into system development positions with larger firms and consulting companies. Prerequisites to the program would include the undergraduate introduction to MIS, the introduction to object oriented programming using Visual Basic and the introduction to COBOL programming. The track would contain the following courses:

1. Application Development using Relational Databases . Potential course content includes the following: a) data modeling, b) database design using IEF, database development using Oracle c) SQL, d) application development using an Oracle database with Powerbuilder, Access, and/or Oracle tools for windows, e) current topics in database management including client-server issues, objected oriented databases and the management of distributed databases.

2. Application Development Related to Legacy Systems. Course content includes the management of COBOL based systems with VSAM, relative files, IDMS and DB/2. Other topics include Micro Focus Workbench tools, test design, metrics and quality control, JCL issues, object oriented COBOL, maintenance of legacy systems, and reverse engineering tools using a tool like Existing Systems Workbench from ViaSoft.

3. Information System Engineering. Course content centers around the development of client-server based systems with client based GUI front ends using IEF. The course concentrates on process modeling, process design, user interface designs (GUI), system generation and system installation.

4. Telecommunications and Networking. Course content includes the design of network systems; network installation, management and maintenance of Novell networks; Internet; Electronic Data Interchange; server hardware, drivers and operating systems; TCP/IP; mail systems and network security systems.

Students would take the database and the application development course related to legacy systems in the Fall semester, and take the information engineering and the

MBA CURRICULUM DESIGN

Weeks in ( )	SUMMER I (5)	SUMMER II (5-7)	FALL (16)	SPRING (16)	PRE (3)
EVERYONE	TOOLS A	TOOLS B	CORE C	CORE D	CORE E
OPTION 1			ACCT	ACCT	
OPTION 2			MIS T1 OR T2	MIS T1 OR T2	
OPTION 3			CUSTOM	CUSTOM	
CORE CONTENT	LEVELING: Strategic Mgt Model, Stat, Econ, Computing (OS, SS, Database, Telecomm)	LEVELING: Acct, Finance, Communications (written, meeting, resume, career) Group Dynamics, Case Training	DECISION MAKING/STRATEGY - Integration, Business Plan, Strategic models, Time Based Competition, TQM, Gobal, Information system requirements	LEADERSHIP - Ethics, Environmental, Training, Leader/Followership, Empowerment, Negotiation	THE FUTURE - Global, Gender issues, Electronic communting, Health care, and other future oriented dynamics
CORE PROCESS	Lecture, self study, problems	2-week MBA Boot camp	Integrated teaching of Core in 6 hour blocks	Integrated teaching of Core in 6 hour blocks	Integrated 3-Week Seminar
TRACK PROCESS	NONE	NONE	Two 3-Hour Courses	Two 3-Hour Courses	NONE
RESOURCE CONSIDERATIONS	Intro by graduate faculty, Tools by summer faculty	Tools by summer faculty, Wrap-up by graduate faculty	Core graduate faculty integration over semester. Evening and weekend potential	Core graduate faculty integration over semester. Evening and weekend potential	
NOTES	Case workshops for faculty	Finance, Modeling, Tools, Capital Budgeting, Cash Mgt, Financial plannning, Currency markets, Stakeholders, Creating Value	Faculty round table, Great books ,Expert resources from industry	Faculty round table, Great books ,Expert resources from industry	

networking course in the Spring semester. IEF will be used in the Fall to do the database design, and it will be used throughout the spring semester to do the process modeling, user interface design, system generation and implementation. The application development course related to legacy systems will evolve as the demand for application developers working with legacy systems changes. Legacy systems will always be around.

MIS TRACK TWO - WORK-GROUP SYSTEM DEVELOPMENT

The second information systems technical track will focus more on the development of systems using object oriented tools, components, and methods. Students in this track will enter the field as system developers and systems support analysts where the environment is primarily newer client-server based systems, targeted at corporate work-groups or smaller environments. Many of these graduates may go to work for smaller companies or into marketing, finance, accounting or production positions where the company wants individuals with a good technical systems background. The prerequisites for this track are the undergraduate introduction to MIS, and an introduction to object oriented programming using Visual Basic.

Students in this track will take the same database course and networking course in the Fall and Spring semesters along with the track one students. The track two students will also take the following two courses in Fall and Spring.

5. Work-Group System Development I. The content of this course includes the development of solutions using tools such as Powerbuilder, Visual Basic, Visual Works, and Access with a client-server database. The course will concentrate on user interface design, prototyping and object oriented system analysis and design.

6. Work-Group System Development II. The content of this course is an extension of the first semester course where the students work in groups on a project for a client in the community. Included in the content of this course will be systems planning, requirements analysis, user interface design, project management, and implementation planning.

PROGRAM MANAGEMENT CONSIDERATIONS

The success of any program depends on good program management. One of the key variables in this program is recruiting students into the program that have the backgrounds to adapt quickly to technically oriented programs, develop the communication and interpersonal skills required, and have the self motivation and drive to hit the ground running. A good student recruitment program requires a good marketing program, visibility with corporations, and a good screening program. Another important dimension is to maintain an active industry advisory group to help evolve and fine tune curriculum development. We have developed a niche program using IEF, so it is important for us to develop a good relationship with companies who use IEF. Maintaining a good alumni system that keeps up with the career paths of our alumni is also essential in maintaining a good relationship with industry.

SUMMARY AND CONCLUSIONS

The MBA with two information system technical tracks is currently in the planning stage. Track one has been in place for a number of years, and has been successful based on the percentage of placement with good organizations. The curriculum for track one has been revised some this past year to an information engineering emphasis using IEF. This is a small niche program that has already been very successful with companies that use IEF. The MBA curriculum design committee is currently considering dropping the general MBA track and having only specialized tracks. The main track would be a technical IS track. This track would also be a niche track that would appeal to specific organizations looking for individuals with a technical MBA. The curriculum content of this track is very preliminary and probably will evolve with input from our advisory committee and from other industry sources. Another source we will use is "IS 95: A Model Curriculum for Information Systems" [8] that is currently in process. As a result of the interest and demand in more technical MBA/MIS programs, the profession may want to consider an effort to start developing curriculum guidelines at the masters level as well.

REFERENCES

1. "Now Comes the Corporate Triathlete", *Business Week*, January 31, 1994, pp. 71-72.

2. "Skill Set Needed for 1996 Graduates" , Groupware Session with Industry Advisory Council., Sept., 1993.

3. "MBA Redesign Committee Report, 1994, College of Business, Northern Arizona University, (Casey Donoho, Chris Lockwood, Alden Lorents, Jon Ozmun)

4. Lorents, Alden C. and Greg Neal, "Information Engineering in the MBA/MIS Program - A Case Study", Proceedings, WDSI, 1994, pp. 601-604.

5. Cohen, Eli Boyd, "Evolving Trends in Information Systems Education", *Journal of Information Systems Education,* Winter, 93-94, pp. 14-17.

6. Dowd, Karen O. and Jeanne Liedtka, "What Corporations Seek in MBA Hires: A Survey" *Selections*, 1994, Darden School, pp. 34-39.

7. Gilbert, Jersey, "The Best Jobs in America", *Money,* March, 1994, pp. 70-73.

8. Longenecker, Herbert E., " IS'95: A Model Curriculum for Information Systems" in process.

9. Sumner, Mary, "A Market Research Study for Graduate Education in Management Information Systems", *Journal of Information Systems Education*, Spring, 93, pp. 9-16.

THE IS UNDERGRADUATE CURRICULUM: CLOSING THE GAP

CHARLES H. MAWHINNEY
JOSEPH S. MORRELL
GERARD J. MORRIS
Department of Computer Information
Systems and Management Science
Metropolitan State College of Denver
P.O. Box 173362, Campus Box 45
Denver, CO 80217-3362
(303)556-3122 mawhinney@mscd.edu

ABSTRACT

Curriculum currency is an important issue in almost all academic disciplines of higher education, but is particularly acute in Information Systems (IS) programs in schools of business. Recent AACSB accreditation guidelines place a strong emphasis on total quality management (TQM) concepts which effectively mandate that schools of business must provide education and other services that address the needs of the local business community. This paper describes the results of a survey of the Metropolitan Denver area business in an attempt to ascertain the curriculum needs for the undergraduate IS major. The preliminary results indicate that (among other things) although the traditional technical knowledge is very important, it may not be as important as more the general abilities of being able to learn, work in teams, and communicate effectively.

INTRODUCTION

The last decade has seen a dramatic change in the educational needs of the information systems professional. Prior to ten years ago, the entry-level position was fairly well defined as a programmer or programmer/analyst in medium to large companies who developed their own applications internally. It was often said that the three most important courses to prepare a graduate for such a position were COBOL, COBOL, and more COBOL. However, during the last decade a diversity of new information technologies, such as desktop computers, database management software, fourth generation languages, networking architecture, and end-user productivity software, have greatly changed the educational needs of the information systems professional. Today we see a multitude of new job titles in this arena that did not exist ten years ago (e.g., database analysts, telecommunications/networking specialists, EDP security/auditing specialists, user support analysts, and office automation specialists).

It is not clear that these technological changes have affected all parts of the country in the same way. Two recent regional studies, one in the southeastern states (Archer, et al, 1993) and one in the metropolitan Chicago area (Knapp, 1993), indicate that the prevailing need is still for COBOL-oriented programmer/analysts. However, we suspect that this may not be true of the region where our school is situated because of possible differences in its industrial base. In our departmental program assessment which was completed during AY 92/93, a survey of our alumni indicated a high degree of criticism for the COBOL orientation of our program. Our assessment also indicated that we should survey the local business community to assess its needs prior to modifying our program.

A recent study in the New England Area (Trauth, et al, 1993) concluded that there is an "expectation gap" between Industry needs and academic preparation. This article alone provides sufficient motivation to foster a tie with industry to work together to close the gap.

The School of Business where this study took place has recently gained "candidacy status" in the AACSB accreditation process. The recently revised philosophy of AACSB accreditation places a very strong emphasis on being "mission driven" (AACSB, 1992). The School has defined its mission as (MSCD, 1993):

WE ARE COMMITTED TO BEING THE LEADER IN THE DEVELOPMENT AND DELIVERY OF HIGH-QUALITY (undergraduate) BUSINESS PROGRAMS THAT MEET THE DIVERSE AND EVOLVING EDUCATIONAL NEEDS OF STUDENTS, BUSINESSES, AND OTHER STAKEHOLDERS (in the six-county service area of Denver, CO).

In essence, the school is a "teaching" institution and its mission makes it necessary to assess the educational needs for its majors in the local community. This survey in the information systems area is the first of what will probably be many that will be undertaken by the school in the near future.

The objective of this survey was to gather information from local business so that:

- The IS major can be reengineered to meet the current needs of local industry using these results combined with other inputs from academia.
- Better curriculum and career advice can be provided to IS majors.
- The marketing of the IS program to prospective students and employers can be improved.
- Better contact with the local business community will be established.

This paper provides a preliminary summary of the data received. The descriptive statistics provide some interesting insights which are expected to be reinforced by the inferential analysis that is currently in process.

RESEARCH METHODOLOGY

THE SAMPLE

Questionnaires were sent to members of three different groups that were expected to account for at least 90% of the information systems positions in the area served by our institution. In each group, the questionnaire was accompanied by a cover letter asking that the questionnaire be forwarded to the person responsible for hiring personnel in information systems/data processing. The first group consisted of the chief executives of the 205 member organizations of the local Chamber of Commerce (Denver). The second group consisted of the 1,460 graduates of our IS program. The third group consisted of 290 members of the local chapter (Denver) of Data Processing Management Association (DPMA). A total of 192 usable responses was received from the three groups.

THE INSTRUMENT

Copies of the questionnaire are available from the authors. It consists of four parts: 1) identification of the respondent, 2) organizational characteristics, 3) respondent's information systems department characteristics, and 4) knowledge areas, covering both business and information system's topics, desired in prospective hires. The bulk of the questionnaire focused on the "knowledge areas" which was based on the DPMA Model Curriculum (DPMA, 1990). The questions regarding "knowledge areas" were structured in a tiered approach starting at a high level (Body of Knowledge, e.g., Programming Languages, Operating Systems, etc.) and progressing to lower levels (Specific Topics, e.g., COBOL, C, DOS, Windows, etc.). Almost all of these items asked the respondent to assess the desired "level of competence" using the following scale (DPMA, 1990):

0 =	None	
1 =	Awareness -	Introductory recall and recognition
2 =	Literacy -	Knowledge of framework and contents
3 =	Concept -	Comprehension as exemplified by translation, extrapolation, and interpretation of meaning
4 =	Detailed Understanding -	Appropriate application of knowledge in a structured or controlled context
5 =	Skilled Use -	Application using analysis, synthesis, and evaluation in new situations

METHOD OF ANALYSIS

The data is being analyzed using the SPSS for Windows statistical package. Frequency distributions and summary statistics have been calculated. Possible relationships between organizational size and type versus the other variables are being explored using contingency table analysis and analysis of variance.

PRELIMINARY RESULTS

The preliminary results are intriguing. Some of the more interesting ones are described here.

Job-Related Observations:

- The most common entry level positions, in order of frequency, are: 1) Programming; 2) Information Center/User Support; and 3) Telecommunications/Networking (see Figure 1).
- Job related experience is a highly desired prerequisite in an overwhelming majority of the responses (see Figure 2).
- Certifications such as CDP rated at a low level of importance.

Topics-Related Observations:

1. Business

- The level of competence required in areas of Accounting, Management, Management Science, and other non-IS areas of business is at a low "literacy" level (see Figure 4).

2. Information Systems (See Table 1)

- Problem solving received one of the highest ratings ("Detailed Understanding" level of competence).
- No programming language rated above the "concept" level of competence.
  - ▲ C, C++, SQL rated at or near the "concept" level in the programming languages while COBOL held it's own among the languages.
  - ▲ Spreadsheet Macros rated well in the programming languages.
- End-User Support, Applications Packages, and Professionalism rated relatively high at the "concept" level.
- No one systems development methodology emerged as a distinct choice and none were rated above "concept" level of competency.
- Database rated between the "Concept" and "Detailed Understanding" level of competence.
- No Database Management System rated above the "literacy" level of competence.
- DOS and Windows were the only operating systems that rated above the "concept" level of competence. None rated at or above "detailed understanding" level of competence.
- No mini, mainframe, or workstation computer platform rated near "concept" level of competence - most were near or below "literacy" level of competence. However, PC platforms rated near the "detail understanding" level of competence.
- Telecommunications and LANs were rated near the "Concept" level of competence but no specific telecommunications/networking topic rated above the "concept" level of competence.

Other results that were not necessarily surprising but were important are: Ability to learn, ability to work in teams, oral and written communication were ranked at the highest level of importance whereas mathematics, natural science, philosophy, and Social Science were ranked much lower in importance (see Figure 3).

It is anticipated that differences will be found in the entry level needs of companies depending on their size and type of business (manufacturing versus service). More specifically, it is anticipated that compared to smaller organizations, the larger companies will place more emphasis on: 1) fundamental conceptual knowledge rather than specific computer skills, and 2) third generation languages rather than 4GLs.

The results from the Telecommunications and Networking part of this survey were analyzed in conjunction with the results of a previous survey of academicians on the importance of Telecommunication. The resulting paper has been submitted to the *MIS Quarterly* special issue on pedagogy and curriculum (Morris, Morrell, Monroe, & Mawhinney, 1994). The instrument was modified to be used to survey IS students in late Spring 1994. Their responses are currently being analyzed. The authors are currently modifying the instrument used in this study to be used in surveying academicians across the country. Those responses will be compared with those described in this paper.

REFERENCES

AACSB. (1992). *Achieving Quality and Continuous Improvement Through Self-Evaluation and Peer Review: Standards for Business and Accounting Accreditation.* American Assembly of Collegiate Schools of Business, St. Louis.

Archer, B. A., Katter, O. E., & Price, B. A. (1993). Are colleges preparing students for computing careers?--a detailed look. *IBS Computing Quarterly, Winter, 16-21.*

Computerworld. (1993). *SURVEY OF IS EXECUTIVES: The DPMA's Model Curriculum for IS Education.*

DPMA. (1990). *Information Systems: The DPMA Model Curriculum for a Four Year Undergraduate Degree.* Data Processing Management Association.

Knapp, J. A. (1993). Information systems: educational offerings vs. industry needs - how well do they match? *Proceedings of the ACM SIGCPR 1993 National Conference*, April 1-3, St. Louis, 18-26.

Lombardo, G. A., & Larsen, G. (1992). Management information systems curricula and courses a national survey. *Proceedings of the Association of Management Annual National Conference*, August 4-7.

Morris, G. H., Morrell, J. S., Monroe, S. R., and Mawhinney, C. H. (1994) Telecommunications in the undergraduate information systems curriculum: you snooze you lose. Metropolitan State College of Denver internal working paper.

MSCD. (1993). *Metropolitan State College of Denver AACSB Strategic Plan.*

Sivitanides, M. P., & Cook, J. R. (1993). The challenge for information systems educators in the 1990s. *Proceedings of the Western DSI*, San Diego.

Source EDP. (1993). *1993 Computer Salary Survey.*

Trauth, E. M., Farwell, D. W., & Lee, D. (1993). The IS expectation gap: Industry expectations versus academic preparation. *MIS Quarterly*, (September), 293-307.

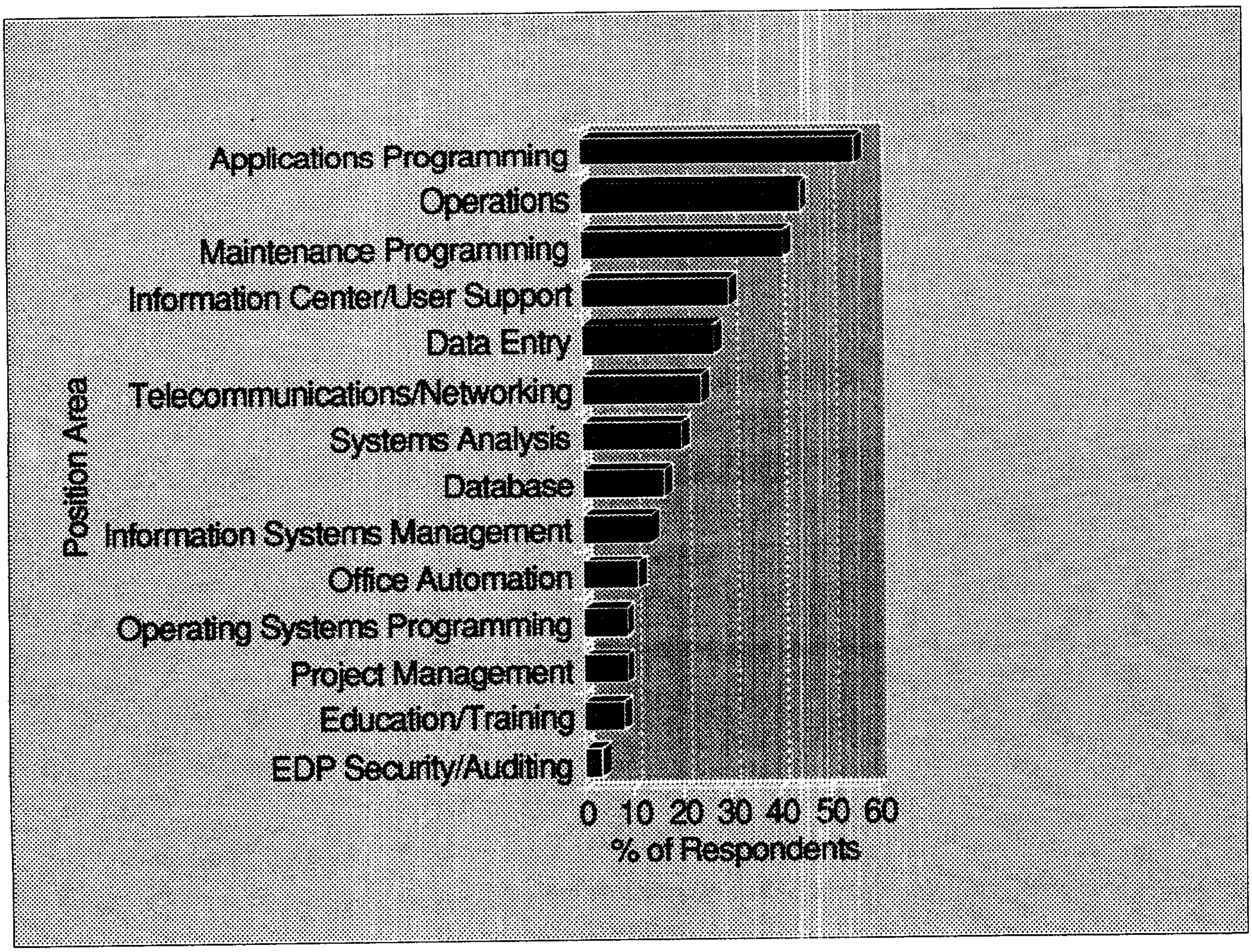


FIGURE 1. Entry level positions may occur in various areas. Please place a check in front of the areas you most frequently fill through new hires.

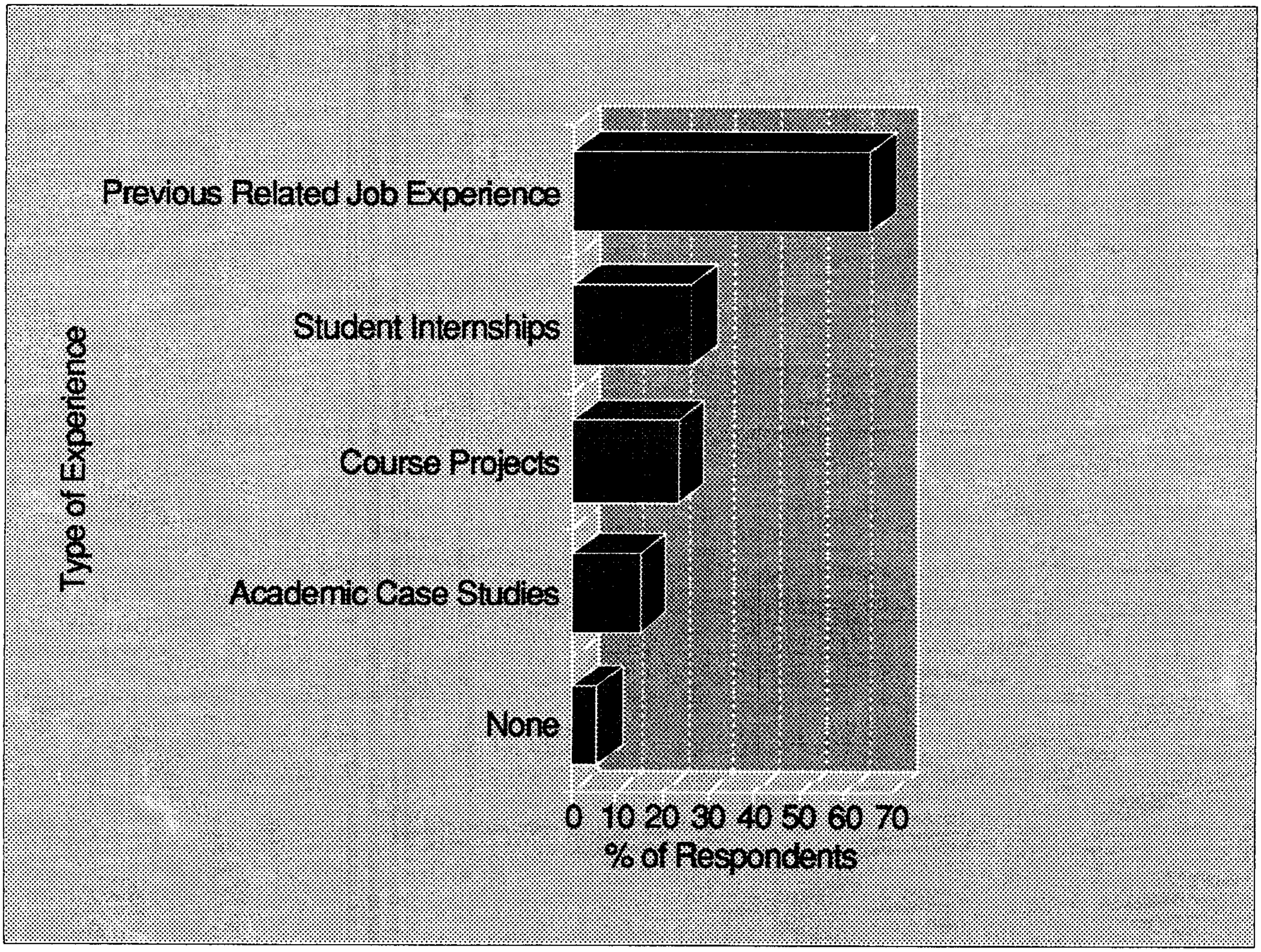


FIGURE 2. Please place a check in front of the level of experience you expect new hires in Information Systems to have.

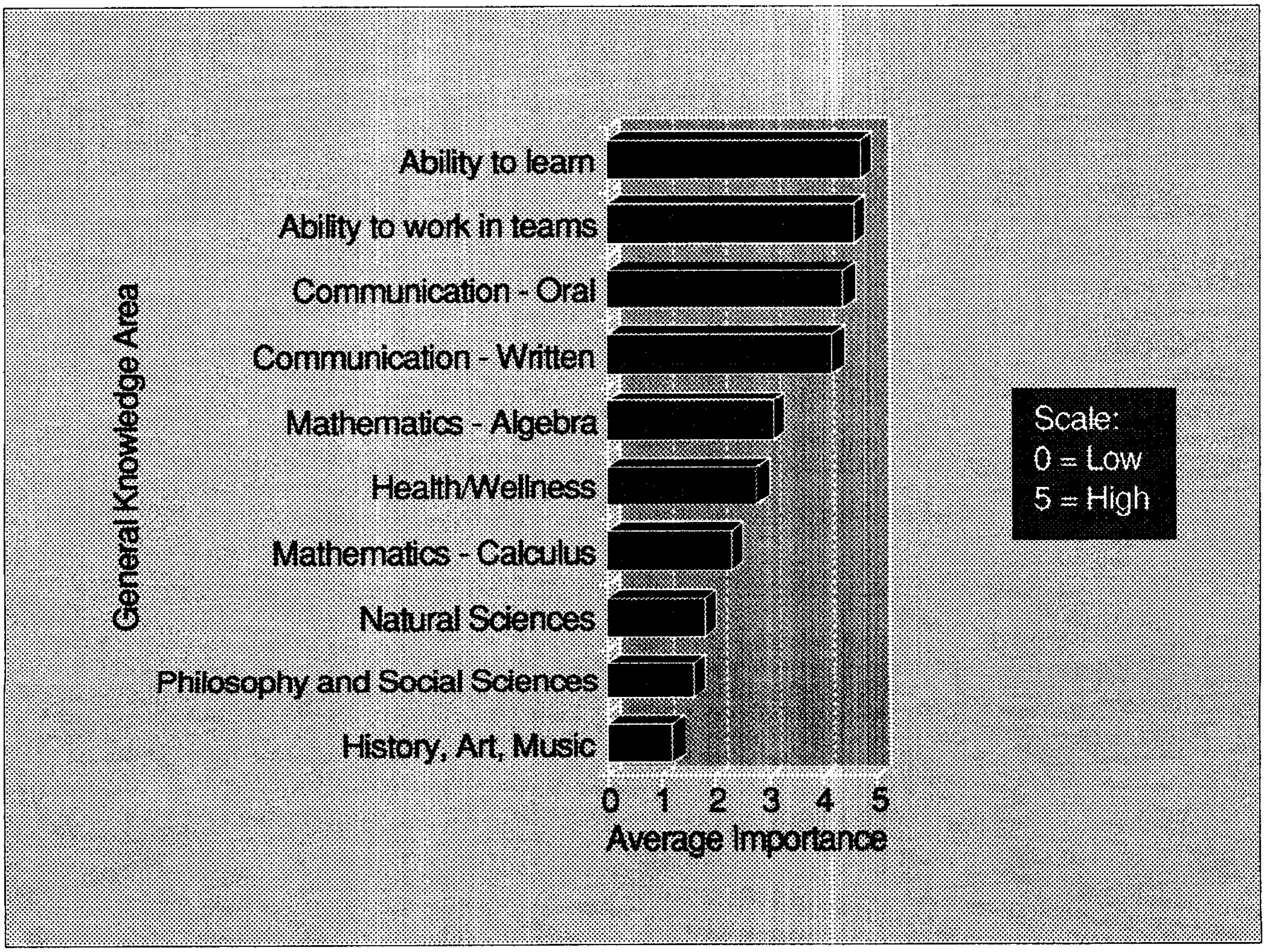


FIGURE 3. How <u>important</u> is it that a new hire be educated in the following general areas? (Circle appropriate response)

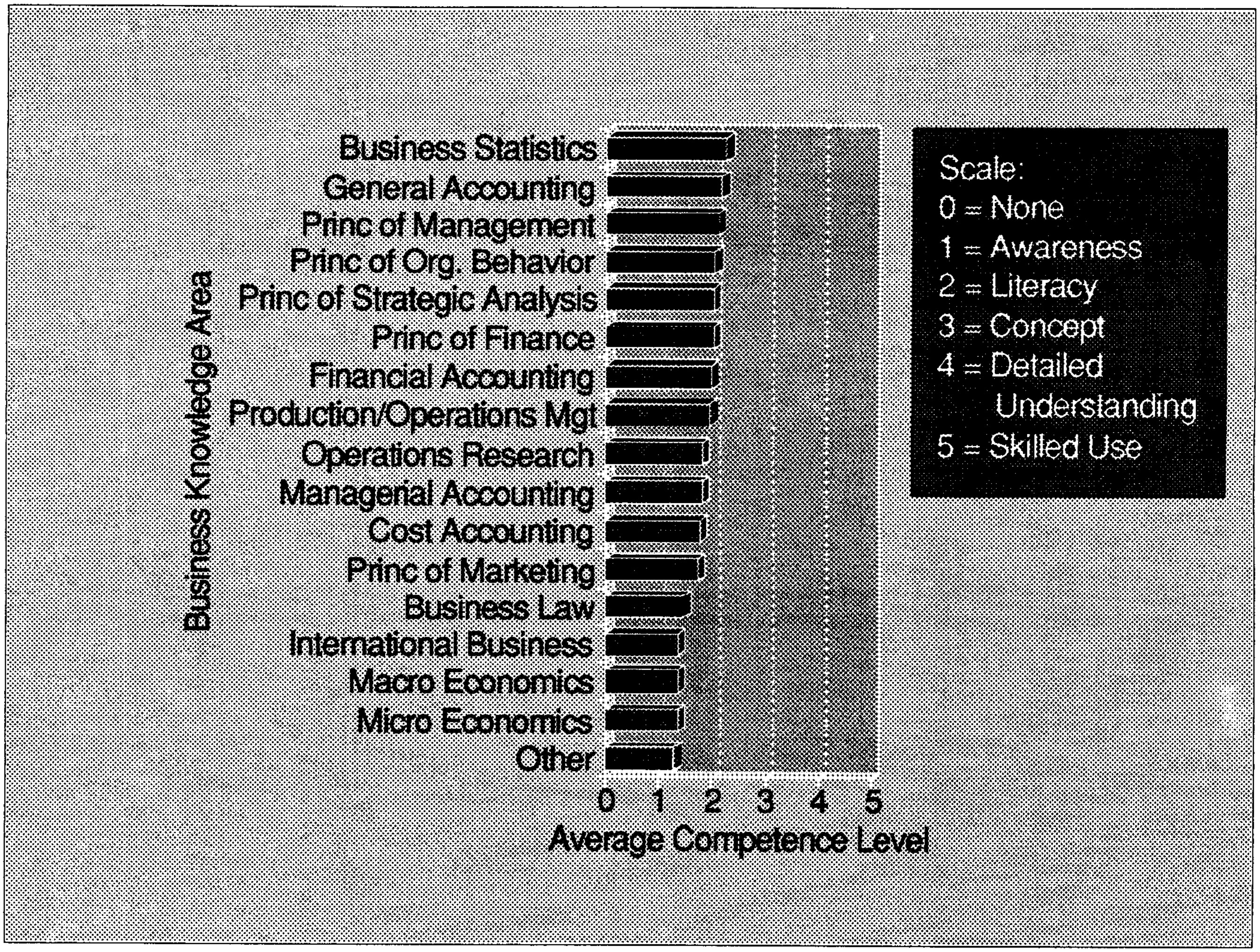


FIGURE 4. For the topics below, indicate the level of competence you expect an entry level employee to have.

TABLE 1. **For the topics below, indicate the level of competence you expect an entry level employee to have.**

Rank	IS Body of Knowledge Area	Average
1	Problem Solving	3.97
2	Implementation & Testing	3.58
3	Professionalism	3.51
4	Database	3.46
5	End-user Support	3.44
6	Systems Development	3.44
7	Computer Architecture/Hardware	3.41
8	Information Retrieval	3.38
9	Development Methodologies	3.34
10	Systems Theory	3.27
11	Algorithms & Data Structure	3.20
12	Nonprocedural Languages (4GL)	3.18
13	Procedureal Languages (3GL)	3.16
14	Operating Systems	3.13
15	IS Design	3.09
16	LANs	3.02
17	Project Management	2.99
18	IS Planning	2.96
19	Telecommunications	2.90
20	Client Server	2.86
21	Change Management	2.82
22	Objected Oriented Languages	2.81
23	WANs	2.60
24	Organization Theory	2.58
25	IS Management	2.52
26	CASE	2.46
27	Decision Theory	2.43
28	Organizational Behavior	2.36
29	Expert Systems/AI	1.88
30	Assembly Languages	1.78
31	Legal	1.57

Scale: 0 = None, 1 = Awareness, 2 = Literacy, 3 = Concept, 4 = Detailed Understanding, 5 = Skilled Use

Hypertext Based Cooperative Help

T.M. Rajkumar
School of Business Administration
Miami University
Oxford, OH-45056
email: rajkumar@miamiu.muohio.edu

Abstract

Drawbacks of hierarchical help and the need for a hypertext based help is discussed. It is also shown how hypertext system can be used to satisfy the architecture for cooperative on-line help. A shell to automate the conversion of a hierarchical help to hypertext based help is discussed. Readability of this hypertext help is presented. Keywords: hypertext,cooperative help, usability, human-computer architecture.

Introduction

Application software and most operating systems such as DOS include some type of on-line help. The inclusion of help in operating systems such as UNIX includes fairly extensive on-line help in addition to reference manuals. In other operating systems the on-line help system is very cryptic such as in DOS 5.0 which gives only an overview of the syntax. Occasionally, operating systems such as DOS 4.0 and earlier versions of DOS have no on-line help at all.

Operating systems such as DOS are widely used by the full spectrum of users from novices to experts. Different types of users have widely varying needs from the help system. Beginners need an explanation of the commands using their own terminology, users experienced with other operating systems need an explanation of unique features, experienced users need a reminder service [9]. To serve all these types of users effectively, help systems must be highly flexible and adaptable to the user.

Many systems such as the VAX/VMS environment provide only a simple hierarchical help structure, and users must access the help by the use of the HELP command. Despite the popularity of hierarchical help systems, several studies have shown that users have difficulty when navigating through menus of several levels. Tombaugh and Mckewen [19] have found that users are likely to make pathway errors (i.e. choose a menu item that does not lead to the desired information) or give up without locating the desired information. Hierarchical help systems are not flexible and adaptable to the different types of users. An answer to this is the use of hypertext help systems.

A major benefit of hypertext is the ability to tailor the presentation to the needs of an individual user [12]. When using hypertext systems, users browse through a text base and make selections according to their needs and interest. Hypertext thus appears to be better suited for on-line applications. Since a large number of current systems provide help through a hierarchical help structure, schemes are required to convert these to hypertext help. Software tools may be applied to help restructure the source text to hypertext text. However intervention may be required during different phases of the conversion process [15]. An objective of this paper is to describe a conversion tool for a hierarchical help system to a hypertext help system. Specifically it will describe the conversion of a help system for DOS 4.0 that was hierarchical to a hypertext based help system under Windows.

Hypertext systems have usability problems that must be solved before they can be put to wide spread use. Help systems can and are used for learning new commands by novice users. This is exemplified by systems such as OS/2 2.0 which come only with on-line manuals. If the system is to be used to learn the commands by the user then the system must be designed to cooperate with the user. This requirement implies that the hypertext system meets user interface requirements. A second objective of this paper is to discuss the requirements for user cooperation in a hypertext based help system. Empirical testing is extremely important for the development of usable computer products, since real users will always interpret some aspect of interface design in other ways than intended by the original designer. Usability testing is done using a reader based documentation measure.

The organization of the paper is as follows: Section 2 reviews help systems and the draw- backs of hierarchical help systems. Section 3 presents the design issues that need to be considered in converting to hypertext help. Section 4 discusses the architecture for coopera- tion and the designed system. The readability of the hypertext help system is presented in section 5. The last section is a conclusion and presents some general guidelines based on the developed system.

Hierarchical Help Systems

The quality of documentation is a universal complaint of both new and experienced users. The problems seems to be worse when the same documents are used by persons with different levels of experience in the environment. An end user needs different information than does an advanced user. The advanced user needs information in a form quite different from that required by a first time user even if both are equally experienced in computer usage [20]. The information delivery problem arises more from the form and mode of presentation [4] than from the information actually presented [20]. In a hierarchical help system a knowledgeable reader is forced to read familiar items before getting to actual material the user is interested in. In addition, a novice user may be introduced to an unknown term before the term is referenced. This places impediments to the learning ability of the novice user. There are 4 reasons people use on-line help documents: a) reminder, b) how to operate a certain command c) find specific facts and d) browse knowledge. Table 1, contains a list of the help required by the various users and the type of help they require [14]. To serve these types of users, a good help system should be able to teach the commands using the information in the help system. The information available should provide the following types of explanative help [9]:

Table I User Needs and Type of Help

User Type	Need	Help Type
Trans-fer	Difference between and new	Tutorial Message
Occass-ional	Reminder How to	Message Tutorial
Expert	Shortcuts, Advanced features	On-line manuals

interpretive explanation: A clearly defined explanation of each command used in the system.

descriptive explanation: Syntax and how the system is to be used with explanation.

graded explanation: Introductory explanation for the novice user and advanced explanation and usage for the experienced user.

reason giving explanation: Explain the need for the command and why such things exist in the environment.

In addition hierarchies can become very deep with many different levels. Deeper hierarchies are more difficult to traverse than broader hierarchies. This is evidenced by the longer search lines. Look ahead help fields have been proposed as a means of removing this problem. However, the addition of look ahead help fields does not consistently improve subject performance and is of minimal practical value in menu navigation [13].

To alleviate these problems hypertext systems have been proposed for the following reasons [12]

- allow for nonlinear helps systems
- allow for different levels of prior knowledge.
- encourage exploration so that the user can learn.
- allow the user to adapt the material to their own learning style.

Hypertext systems allow these changes because they chunk information and interconnect them in a myriad of ways allowing for nonlinear traversal [3]. Users navigate the information in a non linear manner following the interconnections from one piece of information to another. This type of network of concepts can present material to novices so that if they see a terminology they can follow it and find out about it without suffering from learning. Since one order of presentation and degree of detail is rarely optimal for all readers, hypertext based systems help make information available to the varying needs of users.

Considerations for Hypertext Help

Cognitive Issues

Conklin [5] suggests that hypertext environments introduce additional cognitive overhead when compared with more conventional approaches. Readers must engage in decision making as to where to go next and what to read. Another problem with hypertext help systems is the transitional text that helps maintain a sense of materials coherence. The reader may not understand the new sequence and not understand why and how they are relevant. Such disorientation can be reduced with more hierarchical structures, navigational aids and fish eye views [1]. For meaningful learning to take place, pieces of knowledge are subsumed into a hierarchical structure by the reader and must be anchored to existing structures, and integrated with prior material [12]. More dramatic improvements could probably be seen if they are designed as a joint human-machine cognitive system in a cooperative environment.

According to Woods [21],the combination of human-machine cognitive system and effective performance implies maximizing joint performance. Evidence from empirical studies of human advisory systems suggests that an unsatisfactory help or advise was achieved where one human was strongly controlled by another [21]. In contrast, successful advisory sessions had the trademark that both parties were in control of the interaction and contributed to arriving at the right advise. Hence, a good advising system such as an on-line help system must be able to participate in the problem solving process of the user. It must answer questions like: Why is this command not working? What else is needed to make it work? Under what conditions can this command be safely used? How can I prevent such an event from occurring again? Joint cognitive systems, both machine and human, contain partial overlapping expertise that can result in better joint system performance than is possible with either one. Though such joint cognitive (cooperative) systems are defined, not many systems exist. This is principally due to difficulties in achieving cooperative communication between humans and computers that allow for such cooperation.

A major difficulty in achieving such communication has been the lack of an architecture that supports cooperation. Hale, Hurd and Kasper [10] have proposed a layered architecture that supports cooperation by integrating the open systems interconnect (OSI) model for network communication [11] and the Targowski Bowman communication model [18]. This architecture differentiates itself from others in that it emphasizes the nature and scope of referent information required to exchange knowledge as opposed to focusing on the mechanics and linguistics needed to make the exchange. The next subsection discusses the model, specifies the requirements for on-line help, and discusses how hypertext based systems can be used for each of the proposed model layers.

Human-Computer Interface Architecture

The human computer interface architecture matches the application layer of the OSI model with the semantic content/delivery level of the Targowski-Bowman model. The function of this semantic level is to ensure that the right message is sent and the message is correctly interpreted at the receiving end. The layers within this level are i) audience link, ii) session link, iii) environment link, iv) functions and role link, v) symbols link, vi) behavior link, vii) value link and viii) storage retrieval link.

Audience Link: As users become more familiar with the system their knowledge, skills, and requirements from the system change. The audience link level adjusts for this by assessing the recipients level of knowledge and expertise, and adapts the message accordingly in terms of content and form.

In the case of on-line help systems the audience link would adjust for differences in expertise; novice or skilled users. Hence, the system should provide for and distinguish among various users. One merit of hypertext systems is that they can adapt to various users depending on their need. They can present information at varying levels of detail as required by the user.

Session Link: The session link manages the space and time for the message transfer. As a protocol, it decides whether the system or the human is in control for that part of the exchange. As a rule the controlling entity alternates, with each party contributing to the control as needed.

In on-line help systems, this layer distinguishes between who is going to initiate, control what help is displayed and what detail the help is displayed. An implementation of the context sensitive help would be a case of the machine initiating and taking preliminary control. Animating the help or the help screen displaying for a set period of time before displaying the next screen is another example of the machine taking control. A search on a topic, or following links would be a case of the user taking control. Hypertext systems can provide context sensitive help and can also be used for animation and presentation of information.

Environment Link: This layer recognizes that situational factors influence the transmission of knowledge. It recognizes and adjusts for the communication needs during different stages of decision making behavior.

In an on-line help system this translates to finding whether the user is browsing through the help system or is using the help system to find a specific command. It can determine user intentions based on how the help system was triggered. If the help was displayed due to a context sensitive help request, the probability is high that the user was looking for help on a specific command. The system can then adjust and provide the syntax and example. If the user initiated the help at the system level, the environmental link can provide full and detailed help to enable the user to learn about the system. Since hypertext systems have a history feature, they can find any prior help or searches which would indicate it was in browse mode.

Functions and Role Link: This layer accounts for the fact that the function and role of the sender and receiver of the message affects the transmission of knowledge. This adapts for the different roles that are possible and are known and understood by the other party. An on-line help system has to recognize its role as that of a coach, or advisor. As a computer coach, it can unobtrusively help in completing the commands or as an advisor, it can demonstrate through examples. The user has the role of an inactive or an active browser (i.e. the user may have the privilege of placing bookmarks, making annotations or just be in a reading mode). These user roles must be recognized and understood by the help system. Hypertext systems have a minimum 2 modes, those of author and reader. Depending on the mode in which the system was invoked it can ascertain whether the user is an active or inactive browser. Hypertext systems also can provide examples as one of the menu options.

Symbols Link: Effective transmission of knowledge requires that the symbols used by the receiver are understood by the sender and vice versa. The symbol link adapts and compensates for difference in terms of formal languages, jargon, icons used etc.

In a help system symbol links are useful so that information can be presented to different users at different levels of detail. If a terminology is being defined and it introduces or references a term defined elsewhere, a hypertext system can either annotate the information or contain a link. Hence, a hypertext help system can adapt to the needs of the user.

Behavior Link: The actions of the receiver and sender may be interpreted differently based on past experience and assumptions about the sender/receiver. This layer ensures a mechanism for capturing and resolving this information to improve the effectiveness of the transmission of knowledge. Systems that can model the user or learn are examples of behavior link protocols.

In a help system, the behavior of the user can be captured to determine if the user requires command help or pull down menu selection help. From the user history, the help system can determine the user's preference for using the command line vs graphical user interface. Knowledge of user behavior can help the system provide the right kind of help. This knowledge can be obtained based on a) how the help system was invoked and b) Is a mouse system being used or is the user using a combination of key strokes.

Values Link: The values of the sender and receiver have an impact on how the message is received. The value layer accounts for trying to capture the level of computer adaptation of the other party so that the message is correctly received. In a help system, the values of the user can be utilized by the system to provide appropriate help. If the user is a hacker, the help system can be very cryptic in its response to requests for help, thus making the user find the answer. If an end user dreads the use of the computer, the system can be very friendly and provide options at every stage, very forgiving in the mistakes made by the user (multiple levels of undo), and provide help at a great level of detail. Automatic recognition of values is difficult, however the user can be queried by the system prior to entry into the system.

Storage-retrieval Link: The knowledge that is communicated between sender and receiver is affected by past experiences. The storage-retrieval layer enables the capturing of the sender-receiver historical information and ensures that the message is received correctly.

To accomplish this, the help system must maintain the history of all actions of the user. This would be an instantiation of the storage retrieval link. Many hypertext systems allow for the recording of such history. The history can be used to help the system adapt to a novice. For example, when the help system responds by defining a new term and the definition references some other terminology, it can use the history to determine if the user has seen that reference node. If the user has visited the reference node before, it does not display the reference node material. Instead it just provides annotation on the referenced material. History such as the above should not be limited to just this session, but should span previous sessions if possible. History is accumulated in hypertext systems through the use of bookmarks and users annotations.

As can be seen, hypertext systems meet all the requirements of the human computer interface architecture outlined above. The next section discusses a conversion from hierarchical help systems to hypertext help systems.

Hypertext Help Conversion

Text conversion from hierarchical based help systems to hypertext based help systems are feasible, because hypertext systems accept ASCII files as input, and hierarchical help files can be converted to ascii files as needed. This implies that one must structure the interface using the human computer architecture and use the authoring capabilities of the hypertext help systems to identify the links.

Conversion Issues

Document processing to convert from text to hypertext must be automated as much as pos- sible. Involving authors and experts increases not only the time and cost of conversion, but more importantly uses difficult and rare skills [6]. Brown [2] suggests that for a successful implementation the system must be able to capture an existing document and impart struc- ture to it. The software will have to read the source documents, interpret the markups they contain and determine the appropriate format since the screen size is limited.

Raymond and Tompa [17] suggests several impediments to automatic conversion since a structure must be inferred from a careful interpretation. However, Rada [16] specifies that one has to distinguish between clearly-structured and implicitly structured texts. Clearly structured text has obvious structural links, whereas implicitly structured text lacks obvious structure. Hierarchical help texts have obvious structure. Hence, the conversion of hierarchical help is not a problem since the hierarchical structure is apparent.

Raymond and Tompa [17] suggest that the fragmentation required by hypertext may lose semantic information because available linking schemes are too simple to represent the original theme. Text structure can be enforced by introducing levels of abstraction in the document. A term introduced at the highest levels is defined by more concrete terms in the formal theory construction. Watt [20] has found evidence that this structured representation of information improves reader understanding and recall. Garg [7] pointed out that this idea is similar to the concept of aggregation in databases. Abstraction, the mechanism by which information can be stored and retrieved from an information structure at different levels and from a different perspective, also has been found to be useful for maintaining structure.

Conversion Shell

In this research a public domain help system called Sbhelp available for DOS 4.0 was used. Sbhelp used a hierarchical help engine similar to VAX/VMS help facility to display the help for DOS. The help files were all pure ascii files. This help engine was converted to a hypertext help system under Windows 3.0 using Toolbook as the authoring software. The decision to use Windows was simply one of practical need as there were good hypertext systems that had the authoring capabilities.

In order to convert the help system, the help was initially modeled similar to the help system available under Toolbook itself. This was taken deliberately under the assumption that students here see the Windows environment when they enter the laboratory, and hence will be familiar with the Windows interface. However as part of the course under which the hypertext based help was tested, the students would learn a substantial amount of DOS 4. 0. This help facility was targeted at this audience who would need exposure to the help facility. In order to convert to hypertext help, a copy of Toolbook's help structure was used, and the documentation in the help was removed except for just the page structure. Index pages were designed manually for the following subtopics: commands, new terms, summary, batch and config, index of topics. These different groupings were meant for different types of user expertise audience. The new terms and commands were meant for the novice user. The summary, batch, config topics were meant for the more advanced user.

A shell engine was then written using Toolbook's openscript language. The shell engine performed the following tasks:

- It first read the hiearchical help (directory, subdirectories and filenames) and then entered the information in an index under the corresponding index pages.

- It sorted the index page, and then obtained the files and put them one at a time in individual pages following the index. Each page was given a page name corresponding to the filename.

- The shell then scanned the pages for each topic, searching for words referring to other topic names in its corresponding index. If it found a word matching the name of another page it automatically created a link to this page. Such linking was restricted to each of the sub groupings done manually, as linking to every topic would have led to the user later getting lost in hyperspace.

The background for each index was set so that so that a user could click on a topic and display the relevant text. The index for each topic would still be available for the user to see at the left hand side of the screen. This gave the user the capability to navigate to any of the linked pages or choose to go to a totally new topic. The regular hypertext commands such as history, back, bookmarks were available. The navigational aids such as combination keys instead of mouse-based help also were available. An associated example was provided with overt topics in the regular text.

The use of a windows-based help has a drawback when the help is being delivered to a non windows environment. The context sensitive help was not provided, even though the software had the capability to provide it. However this help facility was aimed more as a way of allowing students to learn DOS using hypertext help, rather than a pure on-line help.

The selection of the entire conversion was done automatically except for the initial structuring. This manual initial structure seems to be the norm [6], since the system cannot totally understand the structure by itself and may provide the wrong structure.

Readability of Converted Help

Usability of the hypertext help system is a critical attribute that needs to be carefully tested. Since the above system was designed to help the user learn the material, the comprehension and readability of the system has to be tested. Earlier studies have found in the past that most novice browsers were less systematic and their comprehension of non linear text was not always achieved [12]. Hence, a readability measure of the document was undertaken.

The readability measure used the semantic differential scale of Guillemette [8] designed specifically for hypertext based non-linear text. This scale has the following seven factors: credibility, demonstrative, fitness, personal affect, systematic arrangement, task relevance, understandability. These seven factors explain about 65% of the variance in hypertext documents and have an value of 0.956 [8].

The readability of the hypertext help system was tested on 120 subjects enrolled in an undergraduate end user microcomputer course on operating systems offered by the business school at a large midwestern university. The subjects average age was 19, and had 3 computer courses prior to this course. The subjects rated their knowledge of Windows as 2.6 (scale of 1-5 with 5 representing large knowledge) with a standard deviation of 1.05 out of 5. This was important as the help was in the Windows environment.

The testing procedure was simple. The students were given a brief (5 minutes) introduction to the hypertext help, and were allowed to browse through the help text at their own pace for the next 45 minutes. The students were informed they were to perform a small task at the end. The task was to lookup information about a DOS command in the help and write up some information about it. Care was taken to make it an advanced DOS command that students had not seen earlier in the course. The students were required to record the time it took them to locate the material. This was self reported and not monitored.

The time it took to locate the material was not critical to the success as the readability of the help was the main item being tested. All the subjects were able to locate the material and write up a reasonable explanation of the command. The subjects were asked to fill out the semantic differential questionnaire at the end of the experiment.

Analysis revealed that all seven factors (Credibility, Demonstrative, Relevant, Personal affect, Systematic arrangement, Task relevance, and Understandability) were not evident. Six of the seven factors were identified. Curiously the task relevance factor was missing from the list. It is possible that the subjects assumed that it was relevant since the help was first shown by their instructor.

The students enjoyed using the system. Some typical comments included:

"Good. It is pretty easy to use ."

"I think it is an excellent way to learn about DOS. This is organized a lot better and contains more information than a similar program I have at home."

"Very useful. Need to make improvements in the definitions. Also, [there were] mistakes in the sentence structure." The conversion program did not make any changes to the sentence structures present in the hierarchical help facility.

"The explanations were good and gave me a clear picture of the command. Would be even better if there were examples of the command". The help contained examples which the subject did not notice.

"I found the structure confusing. I go to where I want to be but it is not it. It seems that different sections have same names ? This can be a very useful program."

"Good, but you can sort of get lost in it."

Conclusion

The help system was well received, but still has some of the problems associated with hypertext systems, namely getting lost in the hyperspace. The testing of the system underlines the need for good structure and organization. Automatic conversion systems cannot provide much more structure than is present in the initial hierarchical help system. The designer must provide large amounts of additional structure to be effective. Hence, to design truly cooperative help systems, it is necessary to design the system from scratch and not rely only on automatic conversion systems.

This paper has outlined the needs and requirements for cooperative help systems. The drawbacks and difficulties associated with hierarchical help systems was discussed. An architecture for human-computer cooperation was adapted for help systems. The suitability of hypertext systems and the modifications needed to achieve the cooperation was identified. An automatic conversion system was designed to convert an existing hierarchical system to a hypertext based help system. It has been shown that such systems do not meet all the require- ments for cooperation specified earlier. The study findings indicate that truly cooperative help systems are best designed from scratch as opposed to automatic conversion.

References

[1] Begoray, J.A. "Hypermedia Issues, Systems And Application Areas", *International Journal of Man-Machine Studies*, Vol. 33, 1990, pp. 121-147.

[2] Brown, P. "Integrated Hypertext and Program Understanding Tools," *IBM Systems Journal*, Vol. 30, No. 3, 1991, pp. 363-392.

[3] Campagnoni, F.R. and Ehlrich, K. "Information Retrieval Using a Hypertext-Based Help System", *ACM Transactions on Information Systems*, Vol. 7, No. 3, July 1989, pp. 271- 291.

[4] Cherry, J.M, et al. "Modes of Presentation for On-line help: Full Screen, Split Screen, and Windowed Formats", *Behaviour and Information Technology* 1989, Vol. 8, No. 6, pp. 405-416.

[5] Conklin, J. "Hypertext: An Introduction and Survey", *IEEE Computer*, Vol. 2, No. 9, Sep. 1987, pp. 17-41.

[6] Egan, D.E., et al. "Formative Design-Evaluation of SuperBook," *ACM Transactions on Information Systems*, Vol. 7, No. 1, Jan. 1989, pp. 30-57.

[7] Garg, P.K. "Abstraction Mechanisms in Hypertext", *Communications of the ACM*, Vol. 31, No. 7, July 88, pp. 862-870.

[8] Guillemette, R.A. "Development and Validation of a Reader Based Documentation Measure", *International Journal of Man Machine Studies*, Vol. 30, 1989, pp. 551-574.

[9] Gwei, G. M., and Foxley, E. "Towards a Consultative On-line Help System" *International Journal of Man-Machine Studies*, Vol. 32, 1990, pp. 363-383.

[10] Hale, D.P., Hurd, J.E., Kasper, G.M., "A Knowledge-Exchange Architecture for Collaborative Human-Computer Communication," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 3, May/June 1991, pp. 555-564.

[11] Basic Reference Model for Open Systems Interconnection, *International Standards Organization*, ISO 7498, 1984.

[12] Jones, T. and Schneiderman, B. "Examining Usability for a Training Oriented Hypertext: Can Hyperactivity be Good?", *Electronic Publishing* 1990, pp: 207-225.

[13] Kreigh, R.J., Pesot, J.F., Halcomb, C.G. "An Evaluation of Look-ahead Help Fields on Various Types of Menu Hierarchies", *International Journal of Man-Machine Studies*, Vol. 32, 1990, pp. 649-661.

[14] Nielsen, J. *Hypertext and Hypermedia*, London: Academic Press, 1990.

[15] Rada, R. *Hypertext: from Text to Expertext*, 1991, London: McGraw-Hill.

[16] Rada, R. "Small, Medium, and Large Hypertext", *Information Processing and Management*, Vol. 27, No. 6, pp. 659-677, 1991.

[17] Raymond, D., Tompa, F., "Hypertext and the Oxford English Dictionary", *Communications of the ACM*, Vol. 31, No. 7, July 88, pp. 871-879.

[18] Targowski, A.S., and Bowman, J.P. "The Layer Based, Pragmatic Model of the Communication Process," *The Journal of Business Communication*, Vol. 25, No. 1, Winter 1988, pp. 5-24.

[19] Tombaugh, J., and McEwen, S. "Comparison of two Information Retrieval Methods on Videotex: Tree Structure vs Alphabetical Directory" *Proc. of Human Factors in Computer Systems* pp. 106-110. 1982, New York: Association for Computing Machinery.

[20] Watt, J.H. "Level of Abstraction of Structured Text", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 18, No. 4, 1988, pp. 497-505.

[21] Woods, D. "Cognitive Technologies: The Design of Joint Human-Machine Cognitive Systems", *The AI Magazine*, Vol. 6, No. 4, Winter 1986, pp. 86-92. 18

KNOWLEDGE-BASED AUTOMATED IMAGE INDEXING

Kathleen Hennessey, YouLing Lin, Rama Katragadda, Huitan Lu, Rajasekar Reddy
Institute for Studies of Organizational Automation (ISOA)
College of Business Administration
Texas Tech University.
Lubbock, Texas 79409-2101.
(806) 742-1609, fax -3466 odakh@ttacs.ttu.edu.

ABSTRACT

Ability to integrate and deliver information in the form of text, tabulations, graphics, voice annotations, images, numeric data streams, database query results, documents and forms in timely fashion to support remote decisionmaking has become a survival imperative for many organizations. One of the most intractable problems is automation of indexing of images so they can be retrieved in a variety of contexts. A knowledge-based approach to image understanding developed for the semiconductor industry and for military applications has been extended to formulate indexes from image object descriptors. System facilities, most of which function within 1 second on a 486 PC workstation, include: 1) **knowledge acquisition**; 2) an **image knowledge base**; 3) an **icon generator**; 4) general-purpose **image understanding**; 5) **image compression, indexing and retrieval**; and 6) a **knowledge editor**.

I. Introduction

A. Productivity and information technology. As organizations seek to enter and compete effectively in the global marketplace, the direction of efforts to improve quality and productivity as well as and reduce costs has changed from direct labor productivity to improvement of information and facilities available to those directly engaged in productive tasks. This has taken the form of process automation, integration of decision-making tasks, and provision of information and knowledge to support local decision-making. Consequently, the use of information technology is dramatically increasing and knowledge-base systems are finding wider application throughout the enterprise.

Automated visual facilities are widely used to check placement, fixing and quality/consistency[Savarese,1989]. In addition, higher component densities and smaller dimensions require new developments in precise registration [Moreau, 1988]. Although automation has greatly increased the speed of many visual tasks they still depend largely on human sighting, often through a video monitor, microscope or other visual aid. Because performing visually-based tasks by humans is time-consuming and error-prone, this has become a major factor affecting productivity. When components are complex, lighting is variable, or feature sizes are smaller, it is very difficult to train humans to perform visually-based tasks reliably. Automation of visual tasks is an obvious alternative to the human operator, especially when near-100% accuracy and reliability are required.

B. The challenge: speed, reliability, flexibility, low cost. Automation of visual tasks is largely limited to pixel-by-pixel comparison with reference images, either with reference to a master template or to another image. These methods can be oversensitive to acceptable variations in geometry of objects and to image noise. Use of images of other items in the production line can be insensitive to major objects that occur identically in many components, such as those caused by an out-of-tolerance tool.

Extensive efforts have been made to develop expert systems for automation of visual tasks but these require the skills of a knowledge engineer to interview experienced staff and convert their specialized knowledge into written rules about what the objects to be examined, classified and manipulated look like. These "hard-coded" vision systems are not economically feasible except for very large applications because they have to be re-written for each minor change in object appearance, process, or task criteria. Rapidly changing product lines, markets and ancillary technologies have made such systems difficult to use and prohibitively expensive to maintain; this is also true for many other image-dependent applications such as medical records, law enforcement, currency printing and cytology.

C. Indexing images by contents: an unsolved problem. Visual task operators are often required to make complex judgments about images based on previous experience, such as whether an image "looks like" another image, whether an object in an image is likely to have been caused by a particular process or error, and whether an object is of adequate quality or not. Demand for indexing, compact storage, and fast retrieval of images--particularly so that they may be included in reports--is growing rapidly. Facilities now available with PC-based word processors make it easy to insert images in to text documents; the problem of how to index image contents so that they may be quickly and meaningfully retrieved remains unsolved.

As information technology continues to advance in terms of capacity and communication speed, indexing of images remains an intractable and unresolved problem. A Texas Tech University research team in the Knowledge-based Systems Research Laboratory have devised techniques to extract knowledge directly from images for purposes of classification, indexing and knowledge base development. Using methods such as syntactic representation, fuzzy logic, predicate calculus, feature space mapping and statistical methods for image decomposition, compression, and indexing, the team have developed the basis of a general-purpose image understanding facility available in a set of C libraries.

II. Knowledge-based image analysis: state of the art.

A. Knowledge-based Image Analysis for semiconductor inspection. A set of dynamic facilities for indexing, storage and retrieval of knowledge in images has been developed by a research team in the Knowledge-based Systems Research laboratory at Texas Tech University's Institute for Studies of Organizational Automation. The facilities, originally developed for the demanding sub-micron environment of semiconductor manufacturing, provide a non-contact and non-destructive method of inspection. Because knowledge is acquired directly from images, technical staff in each unit along the production line can teach the system inspection criteria and defect classifications that reflect their particular requirements and concerns. Operators can also change these criteria and classifications at any time without assistance, making it easy to adapt to changing conditions. Images are indexed by production lot, type of device, and the **class of defect the automated vision system found in the image**, as shown in Figure 1.

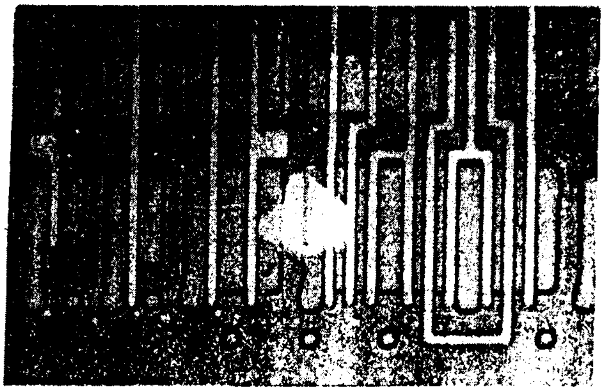


Fig. 1 A semiconductor defect.

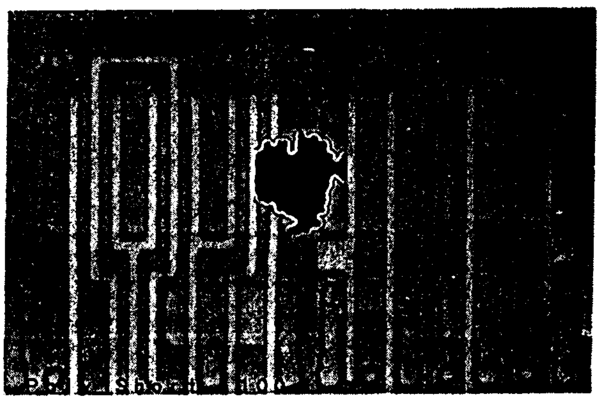


Fig 2. Automatic detection of a defect and classification by operator.

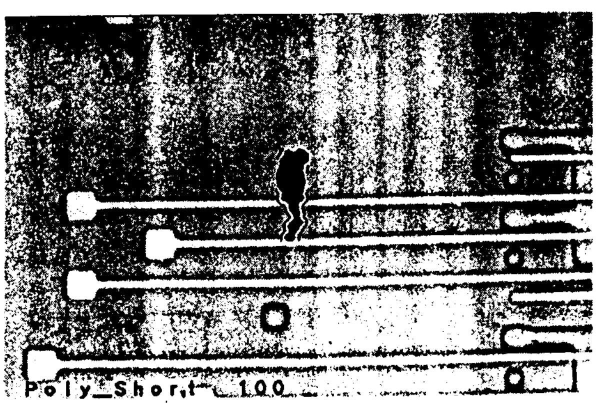


Fig. 3. Automatic detection and classification of another defect.

The research team's knowledge-based image analysis facilities provide a fast, cost-effective means of meeting such requirements They also have been incorporated with other facilities for gross defect detection, fault diagnosis and wafer tracking. Visual classification of a semiconductor defect as an "open" or "short" can later be associated with electrical tests, so that a chip with a faulty component can be rejected as soon as the "killer" defect is visually detected. This means that time-consuming and expensive electrical tests of known defective chips can be skipped, improving throughput. Related knowledge-based image analysis techniques for automated measurement of registration between circuit layers, recognition of distorted wafer identification characters, precise wafer alignment, and monitoring of silicon crystal growth have also been provided to semiconductor manufacturers such as Texas Instruments and their suppliers KLA, Leica, Electroglas, Tencor, and Ultrapointe.

These systems have been installed on the manufacturing floor at Texas Instruments MOS Memory Wafer FAB in Lubbock, Texas and the Silicon Product Department in Sherman, Texas. The technique reduces the cost of image processing equipment by using PC desktop workstations, while at the same time providing speed and accuracy levels equivalent to those of currently available systems costing orders of magnitude more. The technique provides semiconductor manufacturers with flexible, powerful and low-cost inspection facilities using standard PC desktop workstations to diagnose and deal with defects as soon as images of questionable areas are available. In addition to defect classification with an accuracy rate that exceeds 90% after use of only three example images for each type of defect, the system provides fast alignment and dynamic learning facilities.

B. Automated photointerpretation. Human operators performing analysis of military aerial images--photointerpreters--spend hours examining negatives of film taken during reconnaissance flights and digitized images from satellites. Fatigue and lack of expertise can adversely affect reports provided by the PIs, who must also judge whether what they see is significant enough to send a message on communication channels that can be congested just at the time when information in the images is most needed. As reductions in force strength, consolidation of units and redeployment of military personnel continues, it becomes more important to extend the reach of photo-interpreters' expertise and to make selected images quickly available to widely-dispersed operating units. Many of these units only have access to low-bandwidth communications facilities.

The Knowledge-based Systems Research laboratory have developed faciliites for field use on "kneetop"-size PCs; the same software works on the latest high-performance workstations. The system can selectively compress background areas more than specific objects and areas of interest (see Fig. 4), as well as automatically produce the text messages normally prepared by photointerpreters.

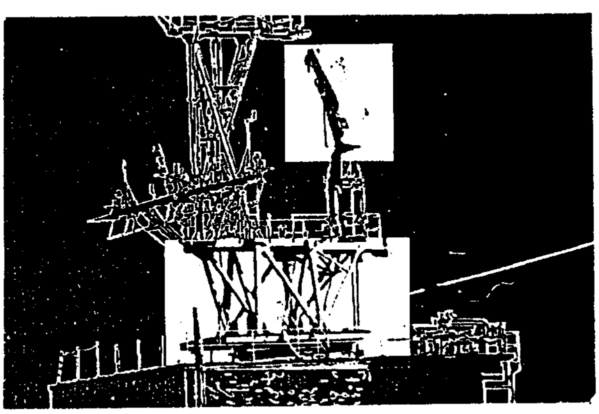


Fig. 4 Multi-level image compression.

Photographs can be automatically indexed by their contents, such as a generic object "aircraft" or a specific object "F-16", as shown in Figure 5.

Retrieval of relevant images from the imagebase is quick, accurate and up-to-date because there is no need to wait for a human operator to identify the objects in each image. It is now possible to retrieve a set of images with a query such as "looks-like [object 15] in [image SJ431-Q] AND is-larger-than [object 8] in [image BR418-L]" that has been generated entirely by mouse clicks.

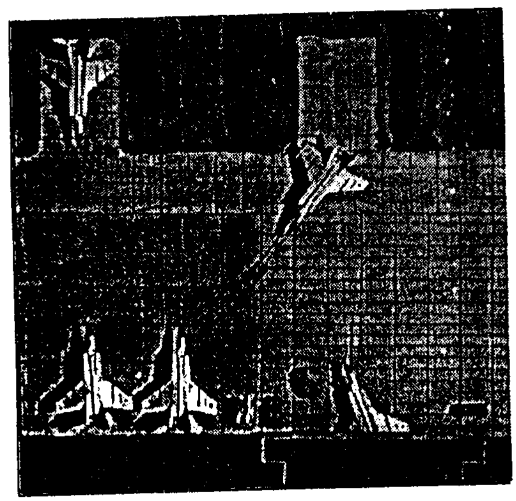


Figure 5a. Image of aircraft.

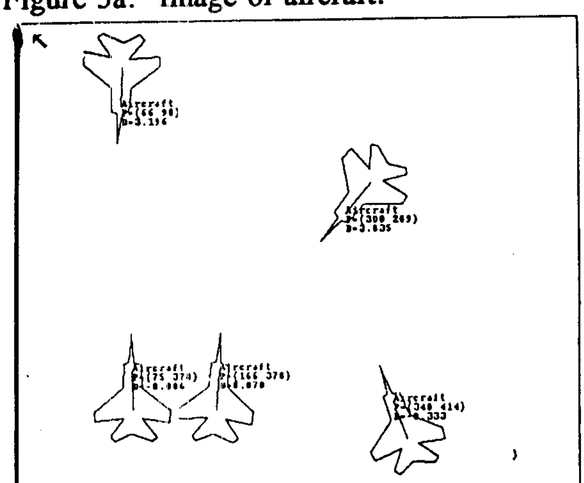


Figure 5b. Automatic indexing of objects.

C. Automated analysis of thin sections of rock. To measure the permeability of rock structures, core samples often knowledge acquisition facility. This problem was tackled by use of predicate calculus, resolving predicate clauses that share the same head predicate as clause groups, involving intermediate image objects with attributes such as common texture or collinearity.

Images of objects can be compressed as much as 600:1 by automatic selection of compression levels for different parts of an image based on their classifications, e.g. current object of interest, background, or other object of interest. These compressed image files are stored in the imagebase; when selected for retrieval, the images are reconstituted and displayed.

D. Representing images with images. It is always possible that retrieval of an image may best be achieved by an operator's use of an icon in a window-style format such as Microsoft Windows or X-Windows. Once the image of an object is indexed, compressed and stored, the system can generate an icon to represent the object. It can also display icons for the operator to select the one that best represents the object, or it can automatically select an icon by matching the descriptors of the object to those of another object that already is repre sented by an icon.

E. Knowledge about objects: dealing with uncertainty. As each of the descriptors is derived, it is assigned a certainty value from which an adaptation of fuzzy logic is used to produce certainty values for objects automatically identified and classified in images. These certainty values are used to determine whether an object or anomaly has been identified correctly.

The system deals with imprecision and uncertainty by incorporating a certainty threshold for each class of object. This is expressed as a radius in feature space within which the descriptors of other objects can be related to one another. When specifications or rules are changed or attributes more precisely defined, the system responds by changing: 1) certainty thresholds; 2) rules for calculation of certainty values, 3) tolerances for measurements specified in the rules, and/or 4) the geometric and/or textual descriptions of objects and their relationships. This provides dynamic updating associated with changes in attribute values and relationships among the attributes, as well as names of image objects.

F. Editing knowledge. One of the main reasons that neural nets have turned out to be so expensive to operate is that, once the net has learned from a large set of examples, it is very difficult for it to "unlearn" or "relearn" information about objects in images. Because the knowledge-based image understanding system usually needs only two or three images to learn the characteristics of an object, and because of the ability of the system to compress an image at selective levels depending on relevance, it is possible to retrieve the set of images used as examples and to add, delete, or change the images in the set, as shown in Figure 7.

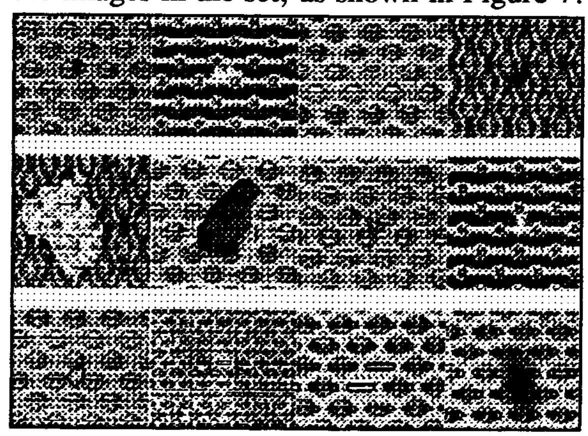


Figure 7. Editing knowledge by adding and deleting images in the knowledge set.

This provides a quick and reliable

costing over a million dollars must be drilled. The samples are then analyzed manually, which can take months. Thin sections made of small pieces of rock--readily available from any drilling operator working in the area--can also be used to obtain the same measures, provided pores found in adjoining thin sections can be connected to give their full dimensions. Research assistants in the Knowledge-based Systems Research laboratory have developed a technique for automated analysis of thin sections of rock, which indexes the images by size and number of pores as well as the identity of the - rock and the layer from which the thin section was taken.

D. Recognition of distorted characters. During manufacture, numbers stamped on components can become almost illegible especially when chemicals, coatings and heat are applied. To meet the needs of quality control efforts, many components or lots are serially numbered so that defects and problems can be traced and the components can be tracked through the manufacturing processes to help identify and determine the causes of delays. Current optical character recognition systems depend upon hardware that cannot be transferred to other equipment and work only on specific industry-standard fonts. The research team have developed knowledge-based optical character recognition facilities that can be taught to read any set of characters. Once installed, the system continues to learn to give the correct identification of characters, even though they are badly distorted. It can learn a new font at any time without being re-installed. In some cases, human operators cannot correctly read these numbers even with a microscope. An example is shown in Figure 6.

III. The problem with image indexing: knowledge about images.

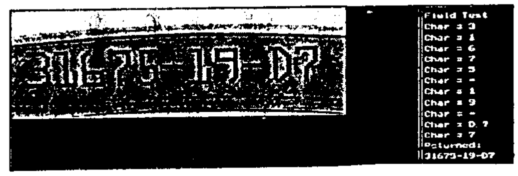


Figure 6. Automatic recognition of distorted characters.

A. Finding objects of interest. Image understanding requirements and problems are especially difficult. For example, in some cases a CAD image is available for basic reference. The image is defined by precise coordinates that provide reference points for objects within a very large field. Objects of interest, which are classified in terms of a range of characteristics such as shape, texture, and size, may occur in any position or rotation within an image; their sharpness and features change with contrast and magnification. Representation of this wide variety of information in a common format is necessary so that it can be cross-indexed and correlated.

The Institute's Knowledge-based Image Analysis System includes the following facilities: 1) **knowledge acquisition** which captures salient features of an image from an operator's interaction with the image and directly from the image itself, converting them to sets of object descriptors; 2) a **knowledge base** containing rule sets derived directly from image descriptors and operators' interaction with one or more example images; 3) an **icon generator** to represent knowledge with icons derived directly from relevant images; 4) general-purpose **image understanding** which identifies and flags anomalies in an image regardless of magnification, rotation or contrast level--usually without use of a prior reference image; 5) **image compression, index-**

ing and retrieval facilities that provide fast retrieval of images from sets of attributed symbols, based on indexed descriptors of the objects they contain; and 6) a **knowledge editor** that allows an operator to view, add, delete and change the set of images from which descriptors are derived for classification and indexing purposes. Most of the facilities function within 1 second on a 486 PC workstation.

Because the system represents images in terms of descriptors, the object-oriented knowledge base can store and process image-based features that are normally difficult to integrate with others. The operator teaches the system to identify an object in an image in one of two ways: 1) teach the system to identify an object or class of objects in an example image by pointing to an example of the object and then naming it; or 2) allow the system to identify areas or objects in the image in which uncharacteristic texture, size, straightness or jaggedness of outline indicate a possible anomaly. These two techniques, along with others, can be used together to produce a powerful image understanding system facility. The image segment within the trained area is analyzed and represented by a set of descriptors which are stored in a knowledge base. The system can then search the knowledge base for other objects whose descriptors approximate those produced for the original object. This provides indexing, fast retrieval and highlighting of relevant images and features. The system has undergone several years of testing in semiconductor manufacturing and other applications.

The architecture of the knowledge-based image understanding system is based on the fact that a portion of an image can be defined by higher-level descriptors such as location, direction, intensity, sharpness etc.; a representation of such descriptors is shown in Figure 6. The image can then be examined or analyzed by a process that is analogous to the action of the human brain by reference to a knowledge-base that contains higher-level image constructs derived automatically from previous experience. The knowledgebase includes ponters to images that have been compressed and stored in the imagebase.

B. Representation of objects in images. Knowledge-based image analysis is based on analysis of images so as to produce descriptors in vector format; values of these descriptors are used to represent objects in images, supported by a knowledge-base. The image of an object, when mapped in feature space and integrated with other objects by statistical and heuristic techniques, can be very compactly and reliably represented. Coordinates are also inherited so that the precise location can be reported when an object is identified. The system can accommodate imprecision, noise and uncertainty and still function within acceptable time constraints--usually less than 1 second response time.

C. Representing, storing and retrieving objects from the imagebase. The knowledge-based image analysis system acquires knowledge from the operator by capturing descriptors of the objects in the image with which the operator interacts. An index argument is constructed from the values the system has provided for each of the object's descriptors. The index argument takes into account the fact that not all descriptors are of equal importance or relevance, and that the certainty level assigned to each of the descriptors affects its significance.

Acquisition, representation and retrieval of knowledge in cases in which operators give the same name (e.g. "particle") to objects that have substantially different visual characterstics is a major obstacle to widespread utility of any automated image way for operators to control the way in which objects in images are indexed.

IV. Conclusions

A dynamic knowledge-based image storage retrieval system has been developed for automated classification of semiconductor defects. The technique acquires knowledge about images from direct operator and system interaction with images, elliminating the need for formulation of rules for identification of objects by operators and/or knowledge engineers. Operators need only be trained to point to and name one reference object to enable the system to identify, index and retrieve specific objects in a series of images. Similar objects, such as characters, in a noisy image can be identified automatically with minimal operator intervention.

Identification and classification of objects involves the capture of distinguishing features of a specific type of object and then using these descriptors to formulate an index argument. Relevance, importance and significance of each descriptor affects the way in which the index item is constructed.

Acknowledgments: Semiconductor inspection facilities were developed under Texas Advanced Technology Program grants 1451, 003644-029, and 003644-143. Knowledge-based image analysis research for photointerpreter automation and support was funded by the U.S. Navy under National Image Display Laboratories Contract No. F105616 and Johns Hopkins University Applied Physics Laboratory Contract 605796-0.

References:

Chang, S.K. and Hsu, Harding, "Image Information Systems: Where Do We Go From Here?," IEEE Transaction on Knowledge and Data Engineering, Vol. 4, No. 5, pp. 431-441, October 1992.

Debenham, John.K., Knowledge Systems Design. Prentice Hall, 1989.

Fu, K.S., Syntactic Methods in Pattern Recognition, Academic Press, N.Y., 1974.

Hennessey, A.K., Hahn, K.S. and Lin, Y.L., "Automated Visual Inspection Using Syntactic Representation of Images," ISOA Working Paper Series, December 1991.

Jarvis, J.R., "Visual Inspection Automation," Computer, Vol. 13, No. 5, pp. 32-38, 1980.

Kosko, B., "Fuzzy Systems as Universal Approximators," First International Conference on Fuzzy Systems, 1992.

Lin, YouLing, "Knowledge-based Image Analysis," Proposal to Texas Advanced Technology Program, Summer 1992.

Moreau, W. M., Semiconductor Lithography: Principles, Practices, and Materials, Plenum Press, New York, 1988.

Savarese, J. "Die Bonding: How to Meet the Challenges of the 1990s," Technical Report to Advanced Mechanization, Inc, 1989.

Yager, R.R. Ovchinnikov, S., Tong, R.M., and Nguyen, H.T.," Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh, A Wiley-Interscience Publication, John Wiley & Sons, Inc., 1987.

You, K.C. and Fu, K.S., "A Syntactic Approach to Shape Recognition using Attributed Grammars," IEEE Transaction of Systems, Man and Cybernetics, Vol. SMC-9, pp. 334-344, June 1990.

TEACHING SYSTEMS ANALYSIS AND DESIGN

PANEL:
Kenneth and Julie Kendall, Rutgers University
Jeff Whitten and Lonnie Bentley, Purdue University
Penny Kendall, Northern Illinois University

The objectives of this panel discussion are to describe strategies for teaching systems analysis and design and to identify new and emerging trends which are affecting the course.

Some of these questions are relevant to teaching the systems course:

1. How has the role of the systems analyst been affected by environmental changes such as business process re-engineering, re-structuring, downsizing, and the transfer of systems analysts into functional business units?

2. What are the most relevant methodologies to be covered in the systems course? Traditional structured systems analysis and design methodologies, using data flow diagrams, logical data dictionary, and structure charts, are still included in most texts. How will the coverage of the traditional structured methodologies be affected by the increasing utilization of information engineering approaches, data-driven methodologies, and rapid application development strategies?

3. How are CASE (computer-assisted software engineering) tools being used in teaching the systems course? How have various CASE tools (INTERSOLV, IEF, etc.) affected the teaching of development approaches? Are courses being re-designed to incorporate the new tools which are a part of the information engineering methodology that is supported by Texas Instruments' IEF? Are CASE tools forcing the re-design of existing courses?

4. What new tools and techniques are being used for teaching the design of screens, reports, and forms. Are new development tools such as Powerbuilder and VisualBasic being integrated into the teaching of systems design concepts?

5. What are the most important skill and knowledge requirements for the future systems analyst? How can these skills be most effectively taught? What new teaching strategies (e.g. case studies, group projects, etc.) are being used to teach the systems course?

Texts:

Kenneth Kendall and Julie Kendall, Systems Analysis and Design. Englewood Cliffs: Prentice-Hall, 1992.

Penny Kendall, Introduction to Systems Analysis and Design. Wm.C. Brown Publishers, 1989.

Jeffrey Whitten, Lonnie Bentley, and Victor Barlow. Systems Analysis and Design Methods. Richard D. Irwin Publishers, 1993.

STANDARDIZATION OF IS PROGRAMS IN NORTH AMERICA

PANELISTS

Oscar Flores
Dept. of Systems Engineering
Universidid Autonoma de Nuevo Leon, Mexico
OFLORES@CCR.DSI.UANL.MX

Jon D. Clark
College of Business
Colorado State University
JCLARK@VINES.COLOSTATE.EDU

Macedonio Alanis
Government of the State of Nuevo Leon
Monterrey, N. L., Mexico
MALANIS@CCR.DSI.UANL.MX

Jose M. Sanchez
Dept. of Artificial Intelligence
ITESM, Mexico
MSANCHEZ@CAMPUS.MTY.ITERM.MX

Carlos A. Gonzalez
Computer Sciences Research Center
Instituto Technologico de Costa Rica, Costa Rica
CGONALV@HURACAN.CR

The increase in business between USA and its neighbors due to NAFTA has created more opportunities for professional people to be involved in the design and implementation of multinational information systems. At the same time, the coordination issues for international operations are more complex today than in the past, therefore requiring a more sophisticated IS infrastructure.

In order to be able to interact or work with people in a multinational company it is necessary to have a common language to facilitate communication. Large multinational companies have adopted English as a companywide standard. Another alternative is to have international exchange programs which include study and work abroad. This will improve the understanding of cultural values and procedures to conduct business in other countries. This type of experiences will reduce the cultural shock when interacting with colleagues from another country.

This panel will discuss these and other issues that justify the standardization of information systems programs in North America.

BUSINESSES' COMPUTING ENVIRONMENT FOR ENTRY-LEVEL PROGRAMMERS AND IMPLICATIONS FOR COURSE CONTENT

Judith C. Brown
Judith C. Simon

Fogelman College of Business and Economics
Management Information Systems and
Decisions Sciences Department
The University of Memphis
Memphis, TN 38152
E-mail: BROWNJ@PROPHET.MEMST.EDU

ABSTRACT

The volatility of the information technology field is such that acasdemic institutions must continually examine and revise course content to stay abreast of industry needs. The purpose of this study was to gather information that could identify any changes that might be needed in course offerings for students who are expected to begin their careers as entry-level programmers. This study attempted to measure the programming language environment and hiring proferences of businesses to ensure that appropriate hardware and software experiences are being provided for students. The results, then, are intended to contribute to the development of an industry-sensitive academic program, thereby assuring students greater opportunity for success when entering the work force.

A questionnaire was sent to area businesses to determine the usage levels of different programming languages and related database packages, along with the projected needs and requirements for entry-level programmers. Related data concerning the computing environment such as hardware and software in use was also collected.

The results of the survey indicate the following needs for classroom instruction.

1. Exposure to distributed processing is desireable as respondents reported heavy microcomputer support for business processing.
2. Students should be given exposure to both DOS and UNIX operating systems.
3. Universities need to seek internships to provide experience for students and then actively market graduates.
4. The market for programmers predicted a slight increase in new hires for 1994.
5. COBOL continues to be widely used, and C appears to be a language firms are increasingly employing.
6. The most widely used database packages were dBase, Foxpro, and Paradox. Most businesses made use of programming capabilities.

HOW TO PLAN AN EMERGING TECHNOLOGIES NIGHT PROGRAM

Ann Robertson
Director of Computing
Mitchell College
437 Pequot Avenue
New London, CT 06320
203-443-2811
E-mail: mitchell@tigger.jvnc.net
Chair, EMTECH-92, Eastern Connecticut Chapter, DPMA

ABSTRACT

This topic is intended to give individuals, departments and chapters of information societies, a step-by-step approach in planning a successful "Emerging Technologies Night Program."

Information professionals from academia (Mitchell College and the U.S. Coast Guard Academy) and from industry (the Eastern Connecticut Chapter of DPMA) came together in June 1992 with objectives to:

- unify the emerging technologies theme of the program by selecting an appropriate presenter for the keynote address.
- determine three separate areas of emerging technologies for which there was most interest.
- provide a program for each track that consisted of an overview, seminars on the latest research/applications, hands-on workshops and vendor booths.
- gather information professionals and students together in meaningful dialog.

The three tracks selected for the November 92 program were:

- Network Computing/the Internet
- Multimedia
- Open Systems

Components that made our program successful and that would be recommended for future planning of similar events include:

- brainstorming
- prioritizing
- planning sessions
- setting agendas
- obtaining facilities
- attracting presenters
- generating funds
- participating and sharing ideas
- evaluating the outcome

TELECOMMUTING: A STRATEGY FOR THE FUTURE OFFICE

Santosh S. Venkatraman &
John W. Rettenmayer

Computer and Office Information Systems
College of Business Administration
Northeast Louisiana University
Monroe, LA 71209-0120
(318) 342-1125

ABSTRACT

During the past few years, corporate America has been experiencing a workstyle revolution, popularly referred to as 'telecommuting'. Telecommuting allows the employee to turn her home into a remote office for her employer. This innovative concept is a product of various social, economic, technological, and environmental factors. In a short span of less than two decades telecommuting has raised big expectations for the office of the future.

Broadly speaking, telecommuting includes any work involving remote data access using a computer and communication channels. Telecommuting offers great flexibility to employees as it allows workers who traditionally work in an office to now work as efficiently in a remote location such as the employee's home. Employers of telecommuters also reap great benefits as they can optimize the allocation of their limited resources, which include people, facilities, equipment and capital.

The purpose of this paper is to provide the reader with a broad view of telecommuting: its advantages, disadvantages, and factors that can help create a successful telecommuting program. We also point out some measures to facilitate the success of telecommuting. In addition to using appropriate computer technology, the success of telecommuting also depends on the willingness of managers to change the way in which they measure and evaluate job performance of employees.

Telecommuting can be a strategic tool for organizations to recruit and retain productive employees, and do so in an efficient and cost-effective manner. When telecommuting is implemented correctly, it should be a "win-win" situation for the employer and the employee. A good telecommuting program is another instance of Information System technology that can significantly enhance the productivity of knowledge workers, in addition to providing a better "work environment."

Teaching the Undergraduate MIS Course

Raymond McLeod, Texas A & M University
Robert Schultheis, Southern Illinois University
James Senn, Georgia State University

The undergraduate MIS course is usually offered at the Junior or Senior Level. In some cases it is preceded by an introductory computer literacy course and in other cases the two courses are combined. The MIS course has never had the primary objective of teaching technology but, rather, information concepts that enable business majors to work with computer systems.

Historical Perspective

In a 1985 survey of AACSB schools (JMIS, Fall, 1985), McLeod found that a wide range of topics were being covered. These topics, and the percentage of schools offering them included: systems analysis and design (88.7), systems theory (85.5), computing equipment (83.9), database (82.3), MIS by management level (79.0), system life cycle (79.0), decision support systems (74.2), database management systems (72.6), MIS by functional area (72.6), programming language (69.4), computer security (64.5), management theory (53.2), data communications (46.8), other (29.0).

In terms of future projections, the 1985 respondents identified several topics that they believed would receive increased emphasis: micro-based MIS designs, database, DSS, end-user computing, networks, 4GLs, and office automation.

Current Emphasis and Future Trends

But the 1985 visionaries could not foresee many topics that are now receiving much attention and appear to be the thrusts of the immediate future: global information systems, ethical issues in information systems, information security, strategic uses of information systems, growing dependence on the communications features of computer systems, applications of information systems to business disciplines, reengineering and integrated technology.

As authors we are in better position than most to see the evolution in the MIS course as we make revisions every two or three years. However, we are no better able to predict future changes than anyone else.

As educators, all of us have a responsibility to not only keep up with change but to influence it in the direction that we think it should take. WE can act as change agents, exploring new concepts and applications and teaching them in the classroom and communicating them to our fellow professionals in industry. The direction that the MIS course takes in the future will be in part a function of where we think business computing should go and how successful industry is in getting there.

FACULTY ADVISORS NETWORK

PANELISTS:

Dr. Longy O. Anyanwu, Chair
Georgia Southwestern College, GA

Dr. Bruce Breeding
Murray State University, KY

Obviously the future of DPMA is in the hands of its youth. Many professionals and DPMA members now see the student chapters as the hub of tomorrow's professional life of the association. Their expected impact on both the profession and the association cannot be over emphasized. Consequently, faculty advisors' responsibilities have taken new dimensions. The Faculty Advisors Network is, therefore, seen by many as a necessary means for effective direction and coordination of the professional activities of the student chapters throughout the entire association. Without a doubt, these activities will later constitute a foundation for their professional life.

The workshop is an interactive discussion between the audience and the panelists. This discussion focuses essentially on the necessity for student chapters on our campuses, student chapter activities, faculty advisors' responsibilities within and without the chapters, and effective implementation of the faculty advisors network for optimal functionality within the association.

SHAREWARE for a BASIC SOFTWARE PORTFOLIO

workshop

John F. Schrage, CSP, Ph.D.
Southern Illinois University at Edwardsville
Management Information Systems Department
Edwardville, IL 62026-1106

The workshop will examine shareware software packages for classroom use. Shareware is an alternative for the educator and students to facility an automated learning environment which will continue to grow through the use of multimedia computer systems. In sending out over fifty letters to shareware authors noting that the writer wanted to distribute the software in a classroom setting, only one shareware author has provided an objection to the procedure. Originally the major source of most of the shareware used has been from acquiring diskettes but now the mode is on CD-ROM diskettes.

Based on the availability of funds in the academic environment for automation, creativity has to be used in obtaining tools to provide both the instructor and the student with realistic materials. As part of my computer courses, students are introduced to a portfolio of public domain/shareware packages to supplement commercial software. A maximizing of costs plays an important part in developing a set of programs to be used. While most commercial computer vendors have a purchasing policy to aid the academic community, a cost is still present which might be too much in some environments.

About ten years ago, a faculty teaching grant was received to acquire software for classroom use. Part of the monies was spent on examining the software realm of public domain and shareware. Since that time, the author has evaluated over 10000 packages which could be used by faculty and students. The majority of the evaluation has been for information systems classes, but with a continuing focus to end user needs. The software portfolio has been used in classes as low as fourth grade through the graduate level setting in business and education.

The three software categories (word processing, spreadsheets, and data base) are used as the base for the portfolio with other categories also examined. Included in the word processing mode is not only the normally word processing software but desktop publishing and writing aids. While many word processors exist, the DOS 5.0 EDIT program makes many shareware writing packages not needed with most word processors able to read ASCII files. Spell checkers and grammar aids are available but normally are not integrated into the word processor.

For the spreadsheet area, a limited set of software exists but add-ons for printing and goal setting, enhanced graphics, and various templates are available. The shareware spreadsheet package, as a competitor of Lotus, has been ASEASY with its graphics being the major factor. The data base provides not only the normal packages but multiple specialized packages. Several data base packages are available in the shareware which emulate the standard of dBASE. Likewise, many sets of data are available for access in the decision making process such as cities, financial data, and census data.

The other area is huge, and that is not even considering the games available which have educational value for a teaching mode. Graphics packages are continuing growing. Clip art is increasing at a tremendous rate based on an increased use of computers in multimedia education. The only problem seems to be the copyright nature of the clips. Even general charting packages, to draw organization charts and logic flows, are available for classes. Simple presentation graphics exist which allow a person to integrate the clip art, drawings, music, and sound.

Many utilities which work in DOS environments have been used and are continuing to be added to the software portfolio. "How-to" software and general tutorial packages seem to be increasing from learning DOS and concepts to chemical reactions. When a student notes a learning problem, the author has been able, in most cases, to find packages to help the student.

Productivity measures have changed from paper and pencil to automated tools. Producing even business cards is a simple task for anyone having a computer and printer. Letterhead stationary, banners, calendars, appointment schedules, and project managers are a few of the work aids noted to classes. Statistical programs to analyze data and generate results are noted for the tool change from paper to computer. Some packages even note what type of statistics test is best for a given criteria. Even some fun packages have been examined which analyze the personality of the individual based on key words.

Almost all programming languages have compilers in the non-commercial area. The FORTRAN compiler found was from Germany and COBOL compilers exist in two modes. Languages such as C, C++, Ada, and BASIC are readily available with tutorials to support the language concepts. Various utilities and sample programs are available for these packages.