ISECON 83

PROCEEDINGS

SECOND ANNUAL

Information Systems Education Conference

Building the Human Resource for Quality Computer Information Systems

March 21- 23, 1983
McCormick Inn
23rd and the Lake
Chicago, Illinois

Sponsored by
DATA PROCESSING MANAGEMENT ASSOCIATION EDUCATION FOUNDATION

ISECON 83

PROCEEDINGS

SECOND ANNUAL

Information Systems Education Conference

Building the Human Resource for Quality Computer Information Systems

March 21- 23, 1983
McCormick Inn
23rd and the Lake
Chicago, Illinois

Sponsored by
DATA PROCESSING MANAGEMENT ASSOCIATION EDUCATION FOUNDATION

THE OBJECTIVES OF ISECON83

The major objective of the annual *ISECON* Conference is to accelerate the development and delivery of education to the present and future work force as rapidly as possible. *ISECON* is intended to be itself a conduit for technology transfer from computer science and engineering, mixed with business and industrial experience, to educators of the work force. The Regents of the DPMA Education Foundation are fully committed and involved with *ISECON*, as a critical instrument in preparing the nation and its data processing managements for the future with a better educated and more effective work force.

The DPMA Education Foundation recognizes the leverage that reliable and relevant data processing can exert on the nation's productivity and economic health. Such data processing, of course, can be no better than the work force that develops and maintains the computer information systems of the nation's enterprises. But that work force now suffers from the rapid growth of an exploding technology, in which few have had basic education and training in the fundamentals of data processing. As a result, the work force lacks important technical skills, standards of excellence rather than mediocrity, and flexibility in its application to national productivity.

The DPMA Education Foundation is concerned with education at two levels:

- for the short run, continuing education for the present work force in seminars, conferences, and short courses
- for the long run, education at the university, trade, and secondary level for the future work force.

One major aspect of education is the creation of reasonable, but challenging expectations, both for the work force and for employers. For example, touch typing is a reasonable expectation never dreamed of a hundred years ago. Writing zero-defect software consistently is a reasonable expectation with today's data processing methodology, and yet the bulk of the work force in the United States regards it as an impossible dream. Therefore, changing attitudes and raising standards of performance are as critical as adding to the skill level in the work force. Since new workers are conditioned in the work place by the performance and limitations of the old workers, it is not good enough to educate new workers only. The expectations and standards must be raised concurrently across the work force.

Fortunately, both the present work force and the new work force can be reached, to a considerable extent, by seminars and conferences aimed at data processing educators and experienced professionals together. Not only are the new methods and attitudes needed equally by both groups, but there is considerable opportunity for learning in the interchange, particularly in educators learning more about needs in the data processing industry.

The sister societies of AFIPS play a key role as sources of technology for education in information systems technology. For example, the structured methodologies now in common use in information system development originated in computer science as structured programming in scientific programming languages; the project management and software configuration techniques in wide use came out of software engineering. Thus, information systems education can expect a continual flow of ideas and techniques from computer science and engineering.

ISECON83 represents a comprehensive look at curriculum proposals of both DPMA and ACM in information systems. We expect these curricula to be an effective force in bringing the benefits of education to present and future practitioners of information systems development in government and business enterprises.

Dr. Harlan D. Mills
Regent
Data Processing Management Association

ISECON 83 PROGRAM COMMITTEE

Program Chairman
M. H. Schwartz
General Manager
Software Quality Service, Inc.

Publications Chairman
Richard C. Linger
Federal Systems Division
IBM Corporation

MEMBERS

Dr. Gerald Hoffman
Manager, Operations Research
Standard Oil (Indiana)

Dr. Herman Hoplin
Associate Professor of MIS
Syracuse University

F. Walter Jenison, Jr.
Chief, Systems Development Service Branch
Tennessee Valley Authority

Dr. H. Albert Napier
Associate Professor MIS
Director of Information
Systems Research Center
University of Houston

Herbert Safford
Staff Administrator
GTE Data Services, Inc.

Celesta Weise
Manager, Dallas Advanced
Education Center
IBM Corporation

Dr. James C. Wetherbe
Associate Professor &
Director, MIS Research Center
University of Minnesota

ISECON 83

PRECONFERENCE WORKSHOPS

Teaching Learning by Design

Dr. Daniela Weinberg
Dr. Gerald M. Weinberg
Weinberg & Weinberg

Teaching the COBOL Sequence

Dr. Robert T. Grauer
Associate Professor
University of Miami

Teaching Information Systems Design

Dr. Nancy Martin
Professor, School of Information Technology
Dr. William McKeeman
Professor, School of Information Technology
Wang Institute of Graduate Studies

Teaching Application Development Without Application Programmers

Professor Perry Sanders
Chairman and Associate Professor of Data Processing
Indiana University, Northwest

ISECON83

March 21-23, 1983
Chicago, Illinois

Table of Contents

EVOLUTION OF INFORMATION SYSTEMS AS AN ACADEMIC DISCIPLINE

Gordon B. Davis
Honeywell Professor of Management Information Systems

University of Minnesota
Minneapolis, Minnesota

Information systems is a relatively new academic discipline. There is considerable diversity in the field and the backgrounds of the 700 or so faculty. It has antecedent elements from other fields, but there are many new elements. It also intersects with a number of academic disciplines.

The domain of information systems may be defined in a number of ways: by elements of a definition, by processes, and by conceptual foundations. There is a necessary technology connection for the field, but its organizational roots suggest it be separate from computer science.

There are good signs of an academic discipline: curricula; scholarly journals; conferences for scholars; and networks for recruiting, promotion, and research. Research is improving. The problem of the future may be maintaining a continuing, productive relationship with information systems practitioners.

Schools of management or business administration are fairly young as academic units within the university. The Harvard Business School is celebrating its 75th anniversary in 1983. The fields of study within business are also young as academic disciplines (except, perhaps, economics). As an illustration, the Accounting Review, the journal that represents the study of accounting as an academic discipline, is only 53 years old. Information systems is the newest of the new. It is going through the ill-defined process by which a new field becomes a legitimate academic discipline. Other fields have done the same, but not recently. This paper will describe the evolution of information systems as an academic discipline, an evolution that began over 15 years ago and is not yet complete. The description is that of a participant because I have been part of it, and I will often refer to things I have done or written because they are the most ready evidence for explaining what I perceive has happened and is happening.

We need to talk about the process by which the field is maturing and how to deal with such comments as the following:

- Information systems is just technology; there is no underlying theory
- There is no research paradigm for information systems
- There are no suitable outlets for good information systems research
- Information systems faculty cannot get promoted because of the lack of an academic discipline

Information systems is challenging because it is at the interface of so many fields and includes a large number of processes and concepts, each of which is interesting in itself. The boundaries of the field have not yet been set, and so people in the field still have the intellectual excitement of explorers.

ACADEMIC DISCIPLINE VERSUS VOCATIONAL TRAINING

It is useful to define "academic discipline" and to distinguish it from courses of study that are generally labelled "vocational." Vocational courses of study are generally thought of as being oriented to the specific requirements and job skills for the immediate, proximate job position. The vocational curriculum is therefore closely matched to current practice. An academic discipline has a longer-term orientation. The course content has a heavy emphasis on underlying principles and concepts for understanding the field and for formulating new answers as the field changes. The content of courses includes material related to current practice, but there is more emphasis on directions and underlying phenomena. There is a feeling that much of specific current practice is best taught on the job.

In an academic discipline, facuty members are evaluated on their command of underlying principles, concepts, and phenomena; this is most frequently demonstrated by research and publications. Both the vocational and the academic discipline approach are valuable and have their place in our educational system. However, within the context of the university, a field will never flourish unless it is an academic discipline.

SOME INDICATORS OF CURRENT ACADEMIC STATUS

There are some facts and observations about the information systems field and the faculty who staff it.

Names, Titles, and Academic Home for Information Systems

The field we are addressing has a number of different academic titles. There is no agreement on what a department or area should be called. I will use the term "information systems," but other titles have had significant use. In a 1979 survey of 124 programs by Nunamaker[17], the titles of programs were as follows:

Number	Name
27	Management Information Systems
18	Information Systems
5	Business Information Systems
4	Computer Information Science
4	Business Data Processing
3	Computer Information Systems
3	Information Processing
3	Information Systems Analysis and Design
3	Information Science
9	Names occurring twice
18	Names occurring once

The location of the information systems academic area within the university is generally in the school of business administration. This is demonstrated by the 87 schools meeting criteria for an information systems program that were identified in the Nunamaker survey.

	Bachelor's	*Master's*	*Total*	*Percent*
Business or Management College	42	25	67	77
Computer Science or Engineering	11	9	20	23

The Information Systems Faculty

There are over 500 faculty members in North America who identify themselves as information systems professors, and there are over 133 schools that have information systems professors. These figures come from the 1983 MISRC/McGraw-Hill Directory of MIS Faculty (DeGross, Davis and Dickson)[7]. Because it is the first directory of its kind, there are faculty who did not get listed and a few schools that are not included. But probably 80 percent of all schools and 70 percent of all faculty are listed. This suggests a population of about 700 information systems faculty.

The distribution among schools for faculty listed in the directory is instructive. Note that the faculty are in the directory because they classified themselves as information systems professors:

Number of Faculty at the School	Number of Schools
1	44
2	17
3	11
4	18
5	9
6	12
7	6
8	5
9	6
10 or more	5

There have been some notable promotion failures for information system assistant professors at some major schools. They didn't get promoted, even though they have a good reputation among their information system academic peers. The question is whether these cases were aberrations or have symptoms of an underlying problem with information systems as an academic discipline.

The Academic Degrees for Information Systems Faculty

There are 36 North American universities (three of them in Canada) that offer a doctorate in information systems as listed in MIS Interrupt. About 30 to 40 doctoral degrees are awarded each year in information systems. However, there are over 100 to 150 unfilled academic positions for information systems faculty.

A characteristic of the population of information system faculty is that well over half of them did not get their primary degree in information systems. Their doctorates are in related fields, and they may have little formal study in the field. There is both good news and bad news in this situation. The faculty is nonhomogeneous in background, and this adds variety to the research and approaches taken; on the other hand, this causes the field to appear to be so diverse as to have no central concepts or set of generally used research paradigms.

The Connection with Information Systems Practice

In a rapidly changing field, there is a tendency to describe current practice rather than teaching underlying principles. There is a feeling of need to have students learn what is happening on the job rather than teaching them principles and underlying knowledge that will aid them in life-long learning. This is an ever-present danger in information systems, and may cause observers to label the field vocational rather than academic.

THE ACADEMIC ANTECEDENTS AND BEGINNINGS OF INFORMATION SYSTEMS

Where were information system principles before there was an information systems field? And when computers and information systems began to emerge, which academic fields taught at least part of the material? The history of the information systems field has been detailed by Dickson[8].

Accounting

The academic antecedents of information systems were the systems courses taught in accounting departments. One might also point to the vocationally-oriented office systems courses, but these generally were taught as vocational job preparation and did not command status as an academic discipline.

A basic decision in the field of accounting as an academic discipline that emerged in the 1930s was the decision to emphasize fundamental principles based on economics and to de-emphasize processing procedures. Under this approach, a student in accounting could graduate without ever studying the processing procedures (the bookkeeping). T-accounts were used as a useful instructional aid, but some textbooks even eliminated this connection to the processing system. In terms of the intellectual development of the field, the decision to teach accounting in a processing-free environment was a wise one. There was periodic discussion in the field that accountants should take a systems course, but the systems course was generally of poor quality and poor intellectual content. It was a survey of business machines, and most students found it easier to learn this on the job. With a good conceptual grasp of the process of accounting, students found little difficulty in learning the accounting systems they found in practice, because the processing procedures were visible and stable and changes were infrequent.

Accounting began with an emphasis on measurement and reporting for external reporting purposes. A recognition that external reports were inadequate for managerial purposes lead to the development of managerial accounting with emphasis on relevant costs. An underlying concept of both managerial and financial accounting is the rational human with unlimited information processing capabilities. Some accountants began to examine this assumption. Their work is termed "behavioral accounting." It is an accepted area of academic accounting research, but it has made little impact in mainline accounting instruction.

Management Science

Operations research, quantitative analysis, and decision sciences were another academic antecedent to information systems. Many of the methods of management science were available before computers, but they became feasible with the advent of computers. Others were developed because of the availability of computers. As a result, modeling, linear programming, sampling, etc., became a part of the tools taught to business students, and these tools require computers to be feasible. Students in marketing, finance, etc., are taught skills of analysis that required computers. Financial databases have become an integral part of the training for financial analysts and business planners.

The Beginnings in the Business School

The first computers were used in business organizations in the United States in 1954 (earlier in Great Britain) for payroll processing. However, the introduction of the IBM 1401 in 1961 followed by the IBM System/360 in 1966 mark the real change in organizational use of computers. The first uses were for well-defined accounting applications, but thoughtful information systems managers soon proposed more comprehensive information systems incorporating management science techniques that would provide better, more timely information for management.

During this time, the instruction that students received in the business schools was oriented to the capabilities of computers and how computers were programmed. There was much emphasis on punched card processing. Students often learned FORTRAN. I wrote my first book, Introduction to Electronic Computers, in 1966 and it had significant emphasis on basic technology plus FORTRAN and a little COBOL. There were workbooks for four major systems: IBM 1401, IBM 7094, IBM 1620, and IBM System/360. The idea that students should learn about the computer in an organizational context rather than in a programming context started to emerge, and my next book, Computer Data Processing, published in 1969, reflected this direction. The field of EDP auditing was defined in a report issued by the American Institute of Certified Public Accountants in 1968, Davis, Auditing and EDP.

The Lack of an Academic Home

Since the natural position of accounting as a provider of information for management was coupled with an instructional concept that de-emphasized the delivery system, accounting did not, in general, provide a leadership role in understanding the implications of the computer technology for information in the firm. The decision sciences had a natural interest in the computer as a part of the system to do analysis for organizational decision making, but this was only a part of the emerging impact of computers on information in the organization.

The First Information Systems Degree Programs

The academic study of information systems began in various schools with one or more faculty members teaching about computers and then formulating some information systems concepts. Perhaps the first major effort associated with a degree was the Management Information Systems major in the Master's degree program in the School of Business at the University of Minnesota. There were three of us who took a leadership role in getting it going: Tom Hoffmann, Gary Dickson, and myself. The starting date was 1968. We obtained curriculum approval and assembled a faculty group from interested faculty members. We had no special funding for anything.

The Management Information Systems Research Center (MISRC) was established at Minnesota because we felt the need for close cooperation with the information systems managers of the large companies in the Twin Cities area. We asked them for more than financial support; we asked for their participation in research and the development of an educational program. The MISRC was the first of several cooperative projects of this type. We hosted the first meeting of the Society for Management Information Systems (now called the Society for Information Management). We enrolled doctoral students and began the process of developing an academic discipline.

THE DOMAIN OF INFORMATION SYSTEMS

The domain of information systems as an academic discipline is defined by the characteristics of the information system, the processes for developing and managing the information system, and the body of knowledge that forms the conceptual foundations. Some of the domain is unique to information systems or uniquely formulated by information systems; other parts of the domain intersect with other disciplines.

Domain Based on the Characteristics of the Information System

In the late 1960s when the field began to emerge, certain concepts were central to the definition of the field. Information was viewed as an organizational resource on a par with land, labor, and capital. Information systems captured, processed, stored, retrieved, and displayed data. Data in a form useful to the recipient was recognized as being information. A frequently cited definition of an information system for an organization (or management information system) is the following:

> "an integrated, man/machine system for providing information to support the operations, management, and decision-making functions in an organization. The system uses computer hardware, manual procedures, management and decision models, and a data base." (Davis)[3]

The elements of the definition describe the domain of the field:

- Man/machine system. The information system is not just computers; it combines information processing capabilities of humans and machines in preparing and using information. The information system, composed of computer technology and human participants, is a system that follows the principles defined by general systems theory. It is also an artifact that can be studied in terms of theories of living systems (Miller)[15].

- Supports operations, management, and decision-making functions in an organization. The context of information systems is organizations and the functions of organizations. Knowledge of the use of information in carrying out operations, managing, and performing decision-making is required in order to design information systems.

- Computer hardware and software. There could be information systems without computers, but computers facilitate much more complex, interesting, and useful information systems. Information systems without computers are to computer-based information systems as oxcarts are to airplanes.

- Manual procedures. The design of human user procedures is vital to the success of information systems. Just as software engineering is a basis for programming, human engineering and knowledge work principles are bases for the design of user procedures for information systems.

- Management and decision models. There are a variety of models that may be used to support decision making. Some may be highly structured and these may be programmed into application systems; other unstructured decisions may be aided by decision support systems.

- Databases. The concept of data as an organizational resource is made operational by databases and database system technology. There may be a computer-based information system without databases and database systems, but the full concept cannot be achieved without them.

Domain Based on Information Systems Development and Management Processes

The information systems development and management processes include both processes that are unique to computer systems and processes that are known to other disciplines that have special formulations for the information processing field.

- Information Systems Planning. The development of the information system strategic plan is an instance of strategic planning, but it has unique characteristics based on the need to derive the information system plan and the information system architecture from the organizational plan. This requires an identification of organizational information needs and requirements. The plan requires consideration of future technology availability and cost. Examples of models for use in strategic planning for the information system of an organization are the Nolan stage hypothesis (best understood as a model of organizational learning) and the McFarlan, McKenney and Pyburn[14] model of strategic choices. Various methods of allocating resources are used to set priorities for application development or enhancement (Bowman, Davis, and Wetherbe)[2].

- Information Requirements Determination. Information requirements must be obtained before an information system plan can be prepared or an information system application can be developed. There are a large number of techniques and methods for information requirements determination that are part of information systems development processes; the selection of techniques in a specific situation is based on a contingency theory for this area (Davis)[4]. The requirements process should also consider the requirements based on individual and organizational behavior and legal and social considerations. There are significant problems in "asking" methods for obtaining requirements, and a comprehensive approach to question formulation is an example of research to improve the user-analyst requirements dialogue (Wetherbe and Davis)[19].

- Application Development and Maintenance Process. Based on information requirements, applications are developed or enhanced. Some of the processes that are part of application development and maintenance are:

- Development Methodologies. Development procedures are used to assure that information system applications meet user requirements. The most common development methodology uses the life cycle as the basis for management, but other alternatives such as prototyping are also recommended in certain types of applications (Naumann, Davis, and McKeen)[16].

- Software Engineering. The body of knowledge defines the processes to design and develop computer programs that produce complete and correct results and are maintainable. This includes the writing of computer programs using various languages.

- Human Interface Design. The design of screens, reports, procedures to effectively interface with humans, so that human capabilities are enhanced and human limitations are reduced.

- Implementation . Implementing systems so that they will be accepted is a difficult task requiring user involvement (Ives and Olson)[12].

- Information Systems Organization and Management. Organizational design, organizational controls, staffing, training, career development, quality assurance, backup, recovery, security, etc., are part of the information system management function.

Domain Based on Conceptual Foundations for Information Systems

The conceptual foundations for information systems are those bodies of knowledge that underlie the design and development of information systems. The conceptual foundations differ from application area knowledge. For example, it may be important to understand accounting (application area knowledge) in order to implement accounting information system applications, but the concept of a system (fundamental conceptual foundation) underlies all applications.

- Concepts of Information. What is information and what are its characteristics: There are several concepts that are useful: the mathematical theory of communication, data reduction, quality of information, and age of information.

- Humans as Information Processors. This area includes humans as information processors, limits on human processing, feedback, effect of data compression, and value of unused information. Much of this relates to cognitive psychology.

- Decision Making Processes. Various models of decision making, methods for deciding among alternatives, and impact of individual differences on decision making.

- Expert Systems. This is a subfield in cognitive psychology and artificial intelligence. Expert systems are applications that model the behavior of experts and thus provide support for non-experts in performing the same tasks.

- System Concepts. Information systems are systems. They are designed using fundamental system concepts of subsystems, control, etc. The application systems, through their life, demonstrate many of the characteristics of living systems in terms of response of systems to stress from the environment.

- Organizational Behavior. The design of organizations and organizational dynamics. These factors affect the design of information systems and the need for information in an organization.

- Concepts of Planning and Control. The organization and operation of these activities and the use of information in performing them. Human behavior affecting planning and control activities.

- Concepts of Economic Behavior (especially in the firm). The economic analysis for decisions.

THE TECHNOLOGY CONNECTION

In defining the domain, computer hardware and software (and related subjects) are important constituents. They represent powerful forces in bringing about the new field of information systems. Without the technology, there would not have been the new field. Even though technology represents only one part of the domain of knowledge for information system, it is a part of the core knowledge. The technology is important because it has set operational boundaries for the field. Pre-computer technology allowed only manual or bookkeeping machine information processing systems, and the boundaries of the field were very small. Not much was conceived of because not much could be implemented. Each new advance in the technology has extended the boundaries of the field because with each advance in technology, the possibilities for the information system have been expanded.

An understanding of the role of the technology in allowing a narrow, relatively uninteresting area to become one of the most exciting and intellectually challenging areas of endeavor is useful in understanding the current interest in the automated office. Office systems have been important but have not been part of the research interests of the universities. With the advent of the personal workstation, electronic mail, etc., the automated office has become a respectable area for research.

THE NECESSITY FOR INFORMATION SYSTEMS SEPARATE FROM COMPUTER SCIENCE

It is obvious that information systems and computer science are related; there is a strong area of intersection. Why shouldn't they be combined into one area -- computer science. The computer science field could have options for further study -- robotics, artificial intelligence, software engineering, information systems, etc. There are good reasons why information systems should not be part of such an arrangement.

The most significant reason for the separation is that information systems is not an extension of computer science; information systems is an extension of the study of organizations, organizational systems, organizational behavior, organizational functions, and management. The nature of the extension is conditioned on the same technology that motivates computer science, but computer science is not an extension of electrical engineering (circuit design); rather, it is an extension of mathematics and algorithmic processes.

Starting from such divergent beginnings, computer science and information systems have different academic cultures. Those from an algorithmic culture have a difficult time relating to organizational issues that cannot be defined with clear and complete models. The "buyer" for the product of the two fields is not the same; appropriate research is not defined in the same way; and approaches to problem solving are different.

The need for a separation of the two fields is really a need for different academic homes, but the intersection of the two fields is sufficiently large to call for significant cooperation and interaction. This does not always happen to the extent desirable.

THE INTERSECTING DOMAINS OF INFORMATION SYSTEMS AND TRADITIONAL ACADEMIC DISCIPLINES

Despite the occasional comment that information systems is technology, the field has an unusually high number of disciplines with which it intersects. The intersection occurs because the information systems area uses knowledge from another discipline as a fundamental part of its body of knowledge. The major disciplines and the intersections are shown in Figure 1 and summarized below:

Field of Knowledge	Intersection with Information Systems
Computer Science	Computer hardware System software Data communications Data management systems Software engineering Application programming
Behavioral Science	Humans as information processors Human behavior in organizational systems Human factors in job design Human resistance to change Behavioral factors in decision making

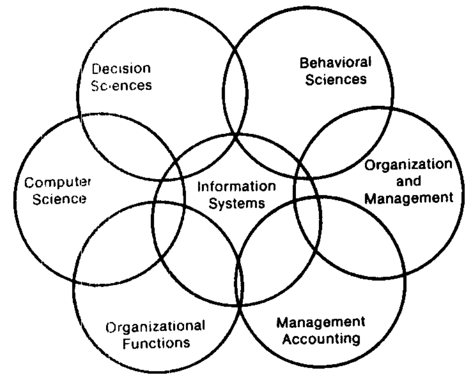


FIGURE 1

INTERSECTION OF DOMAIN OF INFORMATION SYSTEMS WITH OTHER FIELDS OF STUDY AND RESEARCH

Decision Sciences	Decision making Decision models System concepts Information theory
Organization and Management	Organizational design Management style Leadership concepts Supervision concepts Planning and control processes
Organizational Functions	Functional processes in organizations Models of decision making in functional areas
Management Accounting	Cost behavior Management control reporting Performance reporting Variance analysis
Micro Economics	Economic decision-making Pricing Market behavior

SIGNS OF ACADEMIC DISCIPLINE

I have been a participant in much that has happened academically in information systems in the past 15 years, and I have been an observer of the rest. There is little question in my mind but that the field has moved from a loose collection of interested faculty to a fairly cohesive field with good signs of all that it takes to have an academic discipline. There have been reasonable attempts to define curricula; scholarly journals have been established; conferences have been held regularly; "old boy" networks (that include "old girls") have been established; and faculty members are being promoted on the basis of their scholarship in information systems.

Curriculum Recommendations

Curriculum recommendations are an important ingredient in an academic discipline because they indicate reasonable agreement as to the academic content of the field. The earliest curriculum study was the 1972 ACM report of the Teichroew Committee, edited by Ashenhurst (1972) and Couger (1973). Two recent curriculum reports for information systems are significant:

- DMPA Model Curriculum for Undergraduate Computer Information Systems Education, edited by David R. Adams and Thomas H. Athey[1].

- Information System Curriculum Recommendations for the 80s: Undergraduate and Graduate Programs -- A Report of the ACM Curriculum Committee on Information Systems, by Jay F. Nunamaker, Jr., J. Daniel Couger, and Gordon B. Davis[18].

The International Federation for Information Processing (IFIP) also produced an information systems curriculum. It is currently in the final stages of a revision.

Doctoral programs do not tend to be codified in the same way as undergraduate or Master's degree programs, but a reasonable consensus appears in various ways. The doctoral student consortium is an important quality-raising mechanism in many traditional disciplines; a doctoral student consortium is now a regular part of the Conference on Information Systems (with separate funding from the Society for Information Management). There are papers describing the common body of knowledge that might be expected of a person receiving a doctorate in information systems; an example is Davis[6].

Scholarly Journals and a Scholarly Cited Literature

There must be scholarly journals in order to have an academic discipline. The MIS Quarterly is the top ranked journal by information system academics; other desirable outlets for scholarly work are Communications of the ACM, Management Science, a number of other journals published by ACM, and some IFIP and North-Holland journals. The related disciplines also provide publication opportunities, and publication in these related journals demonstrates that the quality of information system research is acceptable to the fields with which it interacts.

Publication in information systems practitioner journals of papers written by academics is to be encouraged in a field such as information systems in which innovation is frequently initiated by practitioners. In the traditional scoring system of academics, such practitioner journals count much less toward promotion. I have argued for more weight in the information systems field. The Harvard Business Review has frequent articles on information systems topics, and I argue a good article there has as much scholarly merit as one in Management Science. Not everyone agrees with me. Datamation has a role in the field that is somewhat unique; it has a wide range of types of articles, but the overall quality is very good. Thus, I encourage publication in it. There are also the journals such as Data Management and Journal of Systems Management that are practitioner journals associated with an information systems organization.

The information systems field has developed some interesting methods for sharing instructional information. Some that I find especially useful are:

MIS Interrupt -- published by the Faculty of Management, The University of Calgary, 2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4.

Computing Newsletter for Schools of Business -- J. Daniel Couger, Editor, published by the College of Business Administration, University of Colorado, Austin Bluffs Parkway, Colorado Springs, Colorado 80933.

Interface -- published by Stephen Mitchell, 116 Royal Oak, Santa Cruz, California 90566.

There are, of course, significant exchanges of information in the major journals cited earlier.

Hamilton and Ives[10] have reported on the knowledge utilization among MIS researchers. They observe that "changes in critical indicators over time reveal a maturation process within the MIS literature but also highlight certain significant barriers to the efficient flow of knowledge." Of interest to me is their identification of four highly cited books and reports:

Anthony, Planning and Control Systems: A Framework for Analysis. Cambridge, MA: Harvard University Press, 1965.

Blumenthal, Management Information Systems: A Framework for Planning and Development. Englewood Cliffs, NJ: Prentice-Hall, 1969.

Davis, Management Information Systems: Conceptual Foundations, Structure, and Development. New York: McGraw-Hill Book Company, 1974.

McKinsey, Unlocking the Computer's Profit Potential. New York: 1968.

These highly cited works (and 15 highly cited journal articles) also provide some insight into the field. Note the management orientation.

Conferences for Information Systems Scholars

There are no lack of conferences for computers and information systems. The National Computer Conference is comprehensive, but those wishing to emphasize information systems can be quickly lost in the crowd. The ACM National Conference is smaller, but only part of the emphasis can be on information systems. The DPMA and SIM conferences are oriented to practitioners. That leaves the Conference on Information Systems (December for the last three years) and special conferences such as ISECON (Information Systems Education Conference). The net result is that information systems academics have places to meet that are relevant to the discipline.

"Old Boy" Networks in Recruiting, Promotion, and Research

There have developed in information systems networks of colleagues who know each other and will respond with truthful information to calls about recruiting and promotion. This type of network also supports the development of strong scholarly activities.

Although there have been some notable failures in promotion of information system academics, a careful study of these cases suggests that the real causes are with the individual schools rather than being a sign of underlying failures. On the other hand, there is still some uncertainty among faculty colleagues about what constitutes good information systems research and where such research is published. This was expressed a few months back when a scholar in marketing commented on a promotion case for a professor who had a reputation in data structures. He didn't understand what the research was about, and the information systems faculty members either explained it so simply that it sounded trivial, or they explained it so that no one could understand it.

CENTRAL TENDENCIES IN INFORMATION SYSTEMS RESEARCH

Some issues related to information systems research are the research paradigms for the field, the models for research, research centers, and the current research interests of the information systems academics.

Research Paradigms for the Field

Research strategies in information systems follow traditional methods. The following is a classification of 331 information systems dissertations written in the period from 1973 through 1979 (Ives, Hamilton, and Davis)[11].

Strategy	Number	Percent
Data-Based Studies:		
Case studies	40	13.9
Field studies	102	30.8
Field test	6	1.8
Laboratory study	45	13.6
Subtotal	199	60.1
Non-Data Studies	101	30.5
Unknown	31	9.4
TOTAL	331	100.0

The research strategies and research paradigms are not new to information systems, but the variety of paradigms accepted by the field is quite large. The one approach that has been suggested that is not in widespread use in other disciplines is action research in which the researcher is a participant rather than a detached observer of the process being studied.

From an academic discipline standpoint, the percentage of non-data studies is too high. These are generally descriptive studies of current practice or descriptions of systems to be developed. Analysis of dissertations subsequent to 1979 indicates the percentage of non-data research is dropping.

There is a need to build a cumulative research tradition, and this is appearing. An example is the set of experiments termed "the Minnesota experiments" (Dickson, Chervany and Senn)[9]. This set has caused a large number of replications and follow-on studies, so that there is a cumulative effect.

Over time, the information systems research community should probably focus on research with information system processes (development, operation, and use) as the dependent variable. Questions about the use process are especially important -- how and why the user is satisfied (participation, satisfaction, and utilization).

Models and Frameworks for Research in Information Systems

Several authors have conceptualized information systems research models. Examples are:

- Mason and Mitroff
- Chervany, Dickson, and Kozar
- Lucas
- Mock
- Gorry and Scott Morton
- Wetherbe and Nolan
- Ives, Hamilton and Davis

These are summarized in Ives, Hamilton and Davis[11], and Wetherbe and Nolan[20]. Two of them illustrate the nature of the research frameworks.

Mason and Mitroff[13] view an information system as:

1. A PERSON of a certain PSYCHOLOGICAL TYPE who
2. faces a PROBLEM
3. within some ORGANIZATIONAL CONTEXT for which he needs
4. EVIDENCE to arrive at a solution, where the evidence is
5. made available through some MODE OF PRESENTATION.

Ives, Hamilton and Davis[11] describe the information system research model as consisting of (Figure 2):

A. Environmental variables
   1. External environment
   2. Organizational environment
   3. User environment
   4. IS development environment
   5. IS operations environment
B. The information subsystem
   1. Content variables
   2. Presentation form variables
   3. Time of presentation variables
C. Process variables
   1. Development process
   2. Operations process
   3. Use process

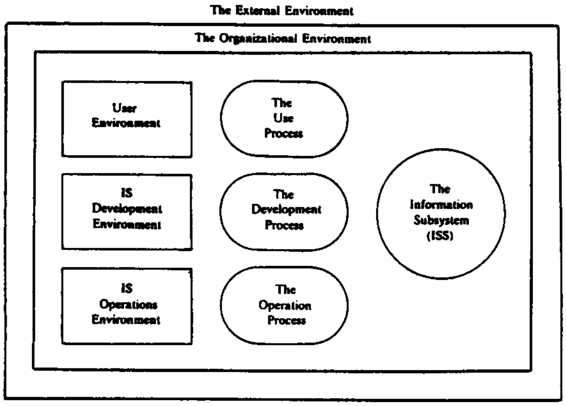


FIGURE 2

A MODEL FOR INFORMATION SYSTEM RESEARCH*

*From Blake Ives, Scott Hamilton, and Gordon B. Davis, "A Framework for Research in Computer-Based Management Information Systems," <u>Management Science</u>, September 1980, Vol. 25, No. 9, page 917.

The research models and frameworks are useful in defining research, specifying variables, and in developing a research tradition.

Research Centers for Information Systems

Starting with Minnesota in 1968, a number of major information systems programs at universities have organized research centers to support information systems research. Following the Minnesota model, most of the research centers have a strong connection with the information systems practitioners. Some examples of the major research centers are:

- Management Information Systems Research Center, University of Minnesota, Graduate School of Management, Minneapolis, Minnesota
- Center for Information Systems Research, Massachusetts Institute of Technology, Sloan School of Management, Cambridge, Massachusetts
- Information Systems Research Associates at McMaster, McMaster University, Faculty of Business, Hamilton, Ontario, Canada
- Center for Information Studies, University of California, Los Angeles, Graduate School of Management, Los Angeles, California
- Center for Research on Information Systems, New York University, Graduate School of Business Administration, New York, New York
- Information Systems Research Center, University of Houston, College of Business Administration, Houston, Texas
- Center for the Study of Data Processing, Washington University, St. Louis, Missouri

Research Interests of Information Systems Faculty

In preparing the 1983 MISRC/McGraw-Hill Directory of MIS Faculty, we asked the 517 faculty who responded to list three research interests (in their own words). An analysis of the terms they used provides some insight into the research interests of the faculty in information systems.

Term Used in a Research Interest	Number Using	Percent Using
System(s)	332	64
Information	161	31
Computer	132	26
Data (including database)	100	19
Analysis	79	15
Design	115	22
Decision support or DSS	123	24
Artificial (intelligence)	13	3
Human	31	6
Behavior	16	3
User	38	7
Control	41	8
Audit	37	7
Security	15	3
Database	78	15
Distributed	26	5
Network	21	4
Office (automation)	40	8
Micro	36	7
Graphic	18	3

SUMMARY AND FUTURE DIRECTIONS

In 15 years, the field of information systems has matured into a reasonable academic discipline. There is still much to be done, but the mechanisms are in place to maintain momentum. The networks, conferences, journals, research frameworks, and research paradigms that help a field to achieve academic excellence and discipline are working. There is still too much descriptive "research" without comparative analysis and research data, but this appears to be changing.

The development of the information systems curriculum is advancing quite well. There is reasonable diversity, but there is also a core of knowledge that most of us agree on. The analysis and design area is still troublesome from an intellectual standpoint. We can describe what people do, but we have not provided the underlying concepts for it.

The main problem I see in the field is to maintain the practitioner connection and at the same time strengthen the field as an academic discipline. The information systems field will become sterile if it loses contact with the processes of its marketplace. I don't think it will happen; but it has happened in other fields. Practitioners and academics need to work toward the happy medium of a solid academic discipline with academics who care about what is happening in practice and practitioners who are interested not only in what works today but also the concepts, theories, and research that make the field so exciting.

REFERENCES CITED IN ARTICLE

If I prepared a full bibliography of references that are relevant to the topic, it would be too long for this article. Instead, because I am reporting on my perceptions of the development of the field, I cite those articles that evidence my participation in the scholarly development. Other related citations made in the article are given here, but there is no attempt to make a comprehensive bibliography.

1. Adams, David R., and Thomas H. Athey, Editors, DPMA Model Curriculum for Undergraduate Computer Information Systems Education. Park Ridge, Illinois: Data Processing Management Association Education Foundation, Committee on Curriculum Development, 1981.

2. Bowman, B., G. B. Davis, and J. Wetherbe, "Modeling for MIS," Datamation, July 1981, pp. 155-164.

3. Davis, Gordon B., Management Information Systems: Conceptual Foundations, Structure and Development. New York: McGraw-Hill Book Company, 1974.

4. Davis, Gordon B., "Strategies for Information Requirements Determination," IBM Systems Journal, 1981, Vol. 21, No. 1, pp. 4-30.

5. Davis, Gordon B., "The Education of Information Systems Educators," Interface, December 1979, Vol. 1, No. 4, pp. 8-11.

6. Davis, Gordon B., "The Knowledge and Skill Requirements for the Doctorate in MIS," Minneapolis: Management Information Systems Research Center Working Paper 81-12 (1981).

7. DeGross, Janice I., Gordon B. Davis, and Gary W. Dickson, 1983 Directory of Management Information Systems Faculty. New York: McGraw-Hill Book Company, 1983. (Available through the Management Information Systems Research Center, 93 Blegen Hall, University of Minnesota, 269 19th Avenue South, Minneapolis, Minnesota 55455. Cost: $10.00)

8. Dickson, Gary W., "Management Information Systems: Evoluation and Status." In M. Yovits, Advances in Computers, New York: Academic Press, 1981, Vol. 20, pp. 1-37.

9. Dickson, G., N. Chervany, and J. Senn, "Research in MIS: The Minnesota Experiments," Management Science, May 1977, Vol. 23, No. 9, pp. 913-923.

10. Hamilton, Scott, and Blake Ives, "Knowledge Utilization among MIS Researchers," MIS Quarterly, December 1982, Vol. 6, No. 4, pp. 61-77.

11. Ives, Blake, Scott Hamilton, and Gordon B. Davis, "A Framework for Research in Computer-Based Management Information Systems," Management Science, September 1980, Vol. 26, No. 9, pp. 910-934.

12. Ives, Blake, and Margrethe H. Olson, "User Involvement in Information Systems Development: A Review of Research," New York: New York University Working Paper, December 1982.

13. Mason, R. O., and I. I. Mitroff, "A Program for Research on Management Information Systems," Management Science, 1973, Vol. 19, No. 5, pp. 475-485.

14. McFarlan, F. Warren, James L. McKenney, and Philip Pyburn, "The Information Archipelago -- Plotting a Course," Harvard Business Review, January-February 1983, pp. 145-156.

15. Miller, James Grier, Living Systems. New York: McGraw-Hill Book Company, 1978.

16. Naumann, J. David, Gordon B. Davis, and James D. McKeen, "Determining Information Requirements: A Contingency Method for Selection of a Requirements Assurance Strategy," The Journal of Systems and Software, 1980, 2, pp. 273-281.

17. Nunamaker, Jay F., Jr., "Educational Programs in Information Systems: A Report of the ACM Curriculum Committee on Information Systems," Communications of the ACM, March 1981, Vol. 24, No. 3, pp. 124-133.

18. Nunamaker, Jay F., Jr., J. Daniel Couger, and Gordon B. Davis, "Information System Curriculum Recommendations for the 80s: Undergraduate and Graduate Programs -- A Report of the ACM Curriculum Committee on Information Systems," Communications of the ACM, November 1982, Vol. 25, No. 11, pp. 781-805.

19. Wetherbe, James C., and Gordon B. Davis, "Developing a Long-Range Information Architecture," Proceedings of the National Computer Conference, Anaheim, California, May 1983, AFIPS Press, Vol. 52.

20. Wetherbe, James C., and Richard L. Nolan, "Toward a Comprehensive Framework for MIS Research," MIS Quarterly, June 1980, Vol. 4, No. 2.

BUSINESS AND GOVERNMENT NEEDS FOR ACADEMICALLY TRAINED INFORMATION SYSTEM DEVELOPERS

James C. Wetherbe
MIS Research Center
University of Minnesota

Al Napier
Information Research Center
University of Houston

A survey of Computer Information Systems (CIS) practitioners in business and government reveals their attitudes about relative value of curriculum topics for academic programs in CIS. Responses are categorized into undergraduate and graduate programs and provide interesting insight into what practitioners value in CIS curriculum.

Computer Information Systems (CIS) programs are rapidly emerging in colleges and universities in response to the intense demand by organizations for CIS professionals. These programs are developing and changing so rapidly that curriculum is far from standardized or stable.

In developing CIS curriculum, the problem is not so much deciding what to include as it is what not to include. CIS is a vast area of study. A problem with developing CIS curriculum is that non-CIS curriculum requirements (at most colleges and universities) leave precious little space for including CIS courses at the undergraduate or graduate level. In such a resource constrained environment, judicious value judgments must be made on what CIS majors should study.

Excellent curriculum studies and recommendations have been prepared by the Association for Computing Machinery (ACM) and Data Processing Management Association (DPMA). The ACM curriculum recommendations have been implemented by many colleges and universities. The DPMA curriculum is designed for undergraduate programs and was published in May of 1981 and is widely implemented.

It goes without saying that as rapidly as information systems technology and organizations are changing, those responsible for CIS curriculum must be continually reevaluating and making adjustments. Accordingly, ongoing curriculum studies are helpful to curriculum evaluation. The idea is not to advocate one study over another. Rather it is to evaluate different studies that provide different perspectives, look for the points of convergence, decide on a niche or strategy for a college or university, and proceed with the best possible judgment.

The purpose of this study was to obtain a purely practitioner view on CIS curriculum from CIS executives of major corporations or government agencies.

As stated earlier, CIS curriculum must be implemented in resource constrained environments. To ask what should be included in a CIS curriculum only results in a curriculum that exceeds resources and therefore requires value judgments to be made by faculty as to what must be excluded.

Therefore, the questionnaire used in this study was designed to force the practitioners to make resource constrained decisions. This was accomplished by having the respondent allocate course content on a percentage basis.

The questionnaire used for the study was distributed to CIS executives at major U.S. corporations and government agencies. Of the 84 respondents, 41 were MIS directors and 12 are in general management. The remaining 31 represent a variety of more specialized management positions in areas such as planning, system development, EDP auditing, and consulting. The industries varied widely. Electronics, auto parts, banking, transportation, paper, petroleum, chemicals, and aerospace firms are represented, as well as large conglomerates. The average current workforce in CIS was 750 and the average monthly hardware expenditure was $705,812. The firms are experienced with computers, as the average number of years using computers was 18.

In the first question, respondents were asked to provide an overall evaluation of the relative amount of emphasis that should be placed on curriculum areas (Figure 1). For undergraduates the primary emphasis is on systems analysis and application programming with hardware/software concepts and database management carrying important weight. At the graduate level, CIS management is the number one topic. Systems analysis and design, decision support systems (DSS), database management systems (DBMS), and data communications and distributed data processing (DDP) all come in with a strong second. Note the low emphasis on applications programming at the graduate level.

Figure 1
GENERAL CIS CURRICULUM

Topic	Bachelor's Degree n = 84	Master's Degree n = 84
Hardware/software concepts	14%	9%
Applications programming	21	6
Decision support systems	9	16
Data communications/ distributed data processing	11	15
Database management systems	14	15
Systems analysis and design	22	16
CIS management	9	23
Total	100%	100%

For undergraduates the primary emphasis is on systems analysis and application programming with hardware/software concepts and database management carrying important weight. At the graduate level MIS management is the number one topic. Systems analysis and design, DSS, DBMS, and data communications and DDP all come in with a strong second. Note the low emphasis on applications programming at the graduate level.

Respondents were also asked to indicate other curriculum areas that were not included in the preceding list that they would like to see included. The most often mentioned additional single topic was communication skills both for undergraduate and graduate programs. Another popular topic was personnel management/ human relations. Each of these topics received mention by about 10% of the respondents. Four topics receiving mention by about 5% were office automation, systems planning, project management, and systems in general.

The second question addressed a specific course in fundamentals of hardware and software concepts. Allocations were as follows:

Figure 2
HARDWARE AND SOFTWARE CONCEPTS

Topic	Bachelor's Degree n = 83	Master's Degree n = 81
Computer architecture & operating systems	19%	16%
Machine & assembly languages	11	8
Programming languages	19	12
Systems analysis and design	16	20
Data communication	14	18
Distributed data processing	13	16
Computers and society	8	10
Total	100%	100%

Not surprisingly, the graduate emphasis is more design-oriented while the undergraduate emphasis deals more with programming. Newer, less understood topics like data communication and distributed data processing have more weight at the graduate level. Computers and society seem to have similar emphasis at each level.

For the bachelor's degree emphasis is on architecture and operating systems and programming languages. System analysis is also important.

There was far less agreement on additional topics. None was suggested which received at least 5% additional support. However, a few were close: data structures, office automation, and software product surveys each received mention by a few respondents.

Application programming received the allocations noted in Figure 3. A few respondents felt that programming was strictly for undergraduates, and therefore left the graduate portion blank. This accounts for the smaller sample size in the second column.

Figure 3
APPLICATION PROGRAMMING

Topic	Bachelor's Degree n = 84	Master's Degree n = 75
APL	5%	6%
COBOL	20	15
FORTRAN	6	5
PL/1	5	5
Pascal	5	6
Job control language	9	8
Structured techniques	16	19
Flowcharting	8	7
Documentation	11	13
Program testing and validity	15	16
Total	100%	100%

Not surprisingly, COBOL received more emphasis than any other language. Based on the allocations undergraduates should spend most of their time in application programming with COBOL. They should also spend significant time with structured techniques and program testing and validity. Graduates should concentrate on structured techniques and program testing, but the amount of time to be devoted to COBOL is still fairly high. Documentation received respectable support on both levels. Interestingly, flowcharting is not "dead" as might be expected.

The only differences that appear significant are those of COBOL and structured techniques. It seems that undergraduates should spend more time studying programming and graduates should concentrate on design issues. The most popular suggestion for additional work was on nonprocedural languages and other high-level productivity tools. About twenty percent of the respondents showed interest in this area at both the graduate and undergraduate levels. Some support was given for studying BASIC at both levels. Mention of control and auditability was also made.

Allocations for system analysis and design follow in Figure 4.

Figure 4
SYSTEMS ANALYSIS AND DESIGN

Topic	Bachelor's Degree n = 82	Master's Degree n = 81
Information requirements analysis	17%	15%
System design	16	12
System development	13	9
System implementation	11	9
System evaluation	8	8
Behavioral issues	7	9
Cost/benefit analysis	8	11
Project management	10	15
Interpersonal skills	10	12
Total	100%	100%

The emphasis at this point seems to be on less technical design issues and more managerial design issues for the graduate student. Undergraduates should spend most of their time on information requirements analysis and system design. Graduates should concentrate on requirements analysis and project management. The heavy allocation for the former suggests that this is indeed a serious issue.

There was very little agreement on additional topics. About 5% of the respondents gave support for communication skills and database design.

Respondents evaluated the database topic as shown in Figure 5.

Figure 5
DATABASE MANAGEMENT SYSTEMS

Topic	Bachelor's Degree n = 83	Master's Degree n = 83
Access techniques	13%	10%
Data structures	18	15
Programming in DBMS environment	16	10
Commercial DBMS packages	13	14
Data security, integrity, privacy	13	16
Database administration	13	19
Data dictionary	14	16
Total	100%	100%

Undergraduates should spend a great deal of their time in understanding data structures, programming, and data dictionaries. Graduates have greatest emphasis in database administration and data security, integrity and privacy. Also, graduates have a fairly heavy emphasis in data dictionaries. The data dictionary allocation is higher than that of access techniques and data structures.

Some additional mention was made of logical data design and information resource management.

Data communications and distributed processing were also evaluated by the respondents. Following in Figure 6 is a summary of allocations for these topics.

Figure 6
DATA COMMUNICATIONS AND
DISTRIBUTED DATA PROCESSING

Topic	Bachelor's Degree n = 83	Master's Degree n = 83
Mini and micro computers	13%	12%
Transmission facilities and networks	16	12
Communication system components	13	10
Networks and controls	13	15
Common carrier services	9	9
Design of networks	15	16
Network management and distributed environment	12	17
Local networks	9	9
Total	100%	100%

Once more, the graduate emphasis is more design-oriented, with network management, design, and controls at the top. Undergraduates should concentrate on transmission facilities, network design, and mini/micro computers with the largest allocation.

There were very few reallocations of this item. Security, integrity, and privacy was the only common addition to this list, with about 5% mentioning this issue.

The question on decision support systems was omitted by about 10% of the respondents. Respondents also reallocated the subject areas less for this area of study. It is certainly the least familiar, and possibly the least tangible of all of the questions in this instrument. Results follow in Figure 7.

It appears that the management decision making is the number one topic for graduate students. Undergraduates should spend more time with DSS analysis and design than other topics. Otherwise, the subject area allocations seem to be fairly evenly spread.

As was mentioned before, very few reallocations were made by the respondents. A few topics were suggested which differed from the above list. These included speaking and writing skills and operational systems (the source of DSS data).

Finally, CIS management was allocated by the respondents. The results are presented in Figure 8.

Figure 7
DECISION SUPPORT SYSTEMS

Topic	Bachelor's Degree n = 76	Master's Degree n = 74
Management decision making	16%	20%
Modeling applications	15	15
Implementation	13	12
DSS software	13	11
DSS analysis/design	18	17
Financial planning languages	12	11
Planning applications	13	14
Total	100%	100%

Figure 8
CIS MANAGEMENT

Topic	Bachelor's Degree n = 84	Master's Degree n = 82
CIS planning	15%	18%
Project management	18	14
CIS organization and staffing	12	13
Installation management	11	10
Hardware/software procurement	11	9
Managing CIS development	14	15
EDP auditing	8	8
Behavioral Issues	11	13
Total	100%	100%

It appears that once again, planning is a leading concern for graduate students. Undergraduates should spend more time with project management.

Communication skills were suggested by 10% of the respondents, and productivity aids were mentioned by about 5%.

CONCLUSION

The survey provides both interesting and useful insight into the type of CIS curriculum most valued by leading CIS practitioners. There are different and appropriate expectations for undergraduate and graduate CIS programs. Undergraduate students need to enter the job market with solid programming skills -- preferably in conjunction with database management systems. They also should have good technical systems design skills.

The product of a graduate CIS program is expected to have a strong management orientation with skills in information requirements, project management, planning, and advanced technology (e.g., DSS, DDP, DBMS).

Both graduates and undergraduates need strong interpersonal skills.

REFERENCES

1. Adams, D.R. (ed.), "Curriculum Recommendations for Undergraduate Programs in Computer Information Systems," DPMA Education Foundation, Park Ridge, IL (1981).

2. Ashenhurst, R.L. (ed.), "Curriculum Recommendations for Graduate Programs in Information Systems," Communications of the ACM (May 1972), pp. 363-398.

3. Couger, J.D. (ed.), "Curriculum Recommendations for Undergraduate Programs in Information Systems," Communications of the ACM (December 1973), pp. 727-749.

4. Wetherbe, J.C. and Napier, A., "MIS Executives Perspectives on MIS Curriculum," Interface (April 1982), pp. 32-35.

THE DPMA MODEL CURRICULUM: AN UPDATE

Thomas I. M. Ho

Purdue University
Department of Computer Technology
West Lafayette, Indiana 47907

ABSTRACT

Since the introduction of the DPMA model curriculum for undergraduate computer information systems education, the focus has changed to the implementation and evaluation of the model curriculum. In order to accomplish these new goals, management of the curriculum development effort has been reorganized. Under the sponsorship of the DPMA Education Foundation, two key committees are pursuing the evaluation and ongoing improvement of the model curriculum.

Coincident with these expected results, several important issues that must be addressed will also be identified.

INTRODUCTION

In 1981, the DPMA Model Curriculum for Undergraduate Computer Information Systems Education[1] was published by the Data Processing Management Association (DPMA). Under the sponsorship of the Education Foundation of DPMA, the model curriculum was developed by the DPMA Education Foundation Committee on Curriculum Development which was chaired by Dr. Thomas H. Athey of the California State Polytechnic University at Pomona.

DPMA EDUCATION FOUNDATION

Mission

The DPMA Education Foundation was established to meet the changing educational requirements of the information processing profession. The Foundation provides leadership in identifying and promoting educational opportunities that will advance the information systems profession.

Objectives

The objectives of the Education Foundation include:

1. To provide educational programs and services for data processing professionals, educators, the computer industry, business, government, or the general public.

2. To assist educational institutions in the development of effective business information processing curricula.

3. To provide funding sufficient to implement these objectives.

Model Curriculum Activities

During 1982, the Education Foundation shifted the focus of the model curriculum effort from publication and awareness to implementation and ongoing development. In line with this change in focus, the Education Foundation established two committees:

1. Curriculum Tracking and Evaluation Committee
2. Curriculum Committee.

The following description of these committees represents the preliminary plans of these committees that are currently being organized. These plans are also subject to the approval and funding support of the DPMA Education Foundation.

TRACKING AND EVALUATION

Purpose

The purpose of the Tracking and Evaluation Committee is to monitor and to evaluate the quality of implementations of the model curriculum in institutions of higher education. The chairman of this committee is Dr. David R. Adams of Northern Kentucky University.

Objectives

The objectives of this committee include:

1. To determine the status of computer information systems (CIS) programs in institutions of higher education.

2. To identify outstanding CIS programs that can serve as role models for other programs that are interested in implementing the model curriculum.

3. To identify a group of curriculum development specialists who can provide lead-

ership to schools that are interested in implementing the model curriculum.

4. To develop guidelines, self-assessment instruments, and/or other evaluation criteria for institutions that have implemented the model curriculum.

5. To sponsor regional conferences and workshops at role model institutions to cultivate model curriculum implementation.

Activities

In order to accomplish these objectives, the committee has planned the following activities for 1983-1984:

1. Survey of CIS education in the United States and possibly Canada by a questionnaire

2. Identification of role model programs
   a. Identification of candidates for role model programs based upon survey results
   b. Selection of role model programs based upon
      (1) responses to self-evaluation instrument sent to role model candidates and
      (2) visits to role model candidate institutions by committee
3. Prototype regional conferences hosted by two role model institutions.

By 1985-1986, the committee intends to establish a network of up to ten geographically distributed role model CIS programs. Generally speaking, the representatives of these role model programs will constitute a cadre of model curriculum specialists who would assist institutions that are developing or improving CIS programs.

ONGOING DEVELOPMENT

Purpose of Curriculum Committee

The purpose of the Curriculum Committee is the thoughtful evolution of the DPMA model curriculum as changes occur in the body of knowledge and in the requirements of society, industry, and academia. The chairman of this committee is Dr. Michael Powers of Illinois State University.

Activities of Curriculum Committee

As before, the continued evolution of the curriculum should again be based on the widest possible input. In addition to questionnaires and one-day workshops that were effectively used in the original development of the curriculum, the committee is concerned about the need to obtain genuinely visionary input into the future of the model curriculum. Therefore, the committee is considering "think-tank" sessions with invited participants and a structured format. These sessions would be 1 to 2 days long and would include a small group of people who represent academia as well as industry.

After conducting these "think-tank" sessions, the committee would organize study groups or task forces with individual responsibility for different parts of the curriculum. This study group organization worked very well in the development of the original curriculum.

OUTSTANDING ISSUES

Several major outstanding issues must be recognized as we attempt to fulfill the promise of the DPMA model curriculum. These issues include:

1. Employer and student credibility for the curriculum
2. Development of instructional materials and aids
3. Faculty development
4. Role of industry in curriculum implementation
5. Institutional evaluation and self-improvement

Employer and Student Credibility

In order for the model curriculum to establish credibility with employers of graduates from programs that adopt the model curriculum, it is imperative that employers know what skills have been mastered by those graduates. In order to recruit thoughtful and capable students into programs that adopt the model curriculum, it is imperative that students know what skills (and performance measures) must be mastered by students in these programs.

Therefore, an instructional plan for the model curriculum must be developed. This plan should reflect equal concern with how we teach as well as what we teach. The foundation of the instructional plan is the instructional design process with three components:

1. Instructional objectives
2. Instructional activities
3. Instructional evaluation.

Instructional Objectives. Instructional objectives serve three purposes:

1. To inform the student about what he should be able to do after instruction

2. To assist the teacher in selecting instructional activities and in determining appropriate evaluation

3. To communicate to others, e.g. prospective employers of graduates, what the instruction is attempting to accomplish.

A recommended format for objectives includes:

1. ACTION - a performance or behavioral verb,
2. CONDITIONS - under which the ACTION will take place,
3. EVALUATION - a statement of acceptable performance.

Instructional Activities. Based upon the instructional objectives, instructional activities are designed to actively involve the student in learning new knowledge and in practicing new skills. A wide variety of alternatives spans a broad spectrum of instructional activities using different media, e.g. lectures, books, or film.

Instructional Evaluation. Instructional evaluation serves several purposes:

1. To measure student achievement, especially to demonstrate mastery

2. To assess the entire instructional process

3. To determine students' attitudes about the instruction

4. To diagnose teaching and learning problems.

Evaluation instruments should test mastery of instructional objectives under the stated conditions according to specific performance measures that define acceptable performance. Again, a wide variety of alternatives spans a broad spectrum of evaluation mechanisms, e.g. examinations, group discussions, and surveys.

Development of Instructional Materials and Aids

It is generally acknowledged that effective instructional materials and teaching aids for CIS are very difficult to find. The model curriculum is intended to encourage the development of instructional materials such as textbooks. However, we must recognize that a curriculum consists of courses. Therefore, instructional materials must be accompanied by instructional aids that have been designed in conjunction with the materials according to an instructional plan such as that previously outlined. Helpful aids should include:

1. Instructor's guide to instructional plan
2. Instructional activities such as:
   a. Lecture outlines
   b. Group discussion strategies
   c. Programming assignments
   d. Case studies
3. Evaluation instruments such as:
   a. Examinations and quizzes
   b. Information system design projects

Of course, such aids are already often included with current textbooks. What is not yet clear is whether such aids were designed in conjunction with those textbooks according to a well-conceived instructional plan so that the total instructional package constitutes a course rather than merely a textbook with ancillary materials.

Therefore, the model curriculum effort must motivate the development of comprehensive instructional plans; i.e. instructional objectives, instructional activities, and evaluation instruments; rather than merely specific course content as has already been accomplished.

Faculty Development

In order to fulfill the instructional plans that will be conceived from the model curriculum, CIS faculty must be developed to learn the instructional models and techniques as well as the CIS knowledge upon which the instruction is based.

With respect to instructional models and techniques, educational institutions must prepare to provide development opportunities, e.g. workshops, for their faculty to learn to become more effective teachers. Furthermore, institutions and CIS academic departments must do more to encourage good teachers with incentives and professional recognition.

Particularly with respect to the CIS knowledge base that is expanding at an accelerating rate, the computing industry must prepare to provide support for professional development of CIS faculty who are already falling dangerously behind the state of the art. The computing industry already provides substantial resources for professional development of information systems non-teaching professionals. In order to sustain the stream of young professionals from CIS educational programs, the computing industry should recognize the necessity for sharing its professional development resources with the CIS teaching profession. Otherwise, our young professionals will continue to learn "history" from outdated faculty in our educational programs.

Role of Industry in Curriculum Implementation

In addition to its indispensable role in faculty development, the computing industry will be the target of gift development programs. For example, the DPMA Education Foundation has mounted a major gifts development program to support the Foundation's plans for grant awards to prepare faculty and to convert institutions to the model curriculum. In addition to the Foundation's goal to become a major conduit for the flow of grant monies from industry and other foundations, educational institutions and academic programs will be pursuing direct aid from industry as well.

No matter how a donor chooses to support deserving efforts to improve CIS education, the computing industry will hopefully recognize that it must be willing to make investments in CIS education in order to continue to enjoy the benefits of a steady stream of competent and well-trained young information systems professionals from CIS academic programs. Users of computing will need these professionals to apply the marvelous hardware and software tools that are provided by the suppliers of computing products. These suppliers will need these professionals to insure the continued growth of the information industry to provide the market opportunities and investment resources that will enable these suppliers to grow and to prosper.

By the same token, academic institutions that solicit and accept the financial assistance of the computing industry must be prepared to hear and to respond to the expectations of industry. It is only natural to expect that industry will expect more from CIS education as a result of industry's role in the improvement of CIS education. If industry is not motivated to offer its viewpoint (a result that this author does not expect), industry's viewpoint should be sought by CIS academic programs. As demonstrated by the model curriculum development effort thus far, industry's input makes a substantial contribution to the effectiveness of education in applications-oriented disciplines such as CIS.

Institutional Evaluation and Self-improvement

In order to evaluate and to improve the implementation of the model curriculum in CIS academic programs, the following aspects of CIS programs should be investigated:

1. Curriculum
2. Faculty
3. Facilities
4. Placement.

Whether these investigations should be performed as part of an academic program accreditation process is still an open question. Certainly, accreditation can be viewed as a mechanism to assist academic programs to acquire the resources that they need so desperately.

Even without the opportunity for accreditation, CIS academic programs should continue to evaluate their progress. To assist these programs in their evaluation, self-assessment instruments could be designed for their use. Even without accreditation standards, responses to these self-assessment instruments could be compared against the opinions of CIS educators and practitioners who have been invited from outside an institution to examine its CIS program.

CONCLUSION

The DPMA model curriculum for undergraduate computer information systems education is already an important achievement. However, it has only begun to affect CIS education in ways that it was expected to make a difference. Furthermore, it has the potential to make a difference for CIS education and for the information industry in ways that are only beginning to become apparent. In order to fulfill its potential, the model curriculum effort will need the broad participation and support of all segments of its constituent community: educators as well as practitioners from both the supplier and the consumer sectors of the information industry.

REFERENCES

1. Adams, David R. and Athey, Thomas H., editors. DPMA Model Curriculum for Undergraduate Computer Information Systems Education. DPMA Education Foundation, 1981. Available at a cost of $4 from:
   DPMA Education Foundation
   505 Busse Highway
   Park Ridge, IL 60068

A COMPARATIVE ANALYSIS OF INFORMATION SYSTEMS CURRICULA

William W. Cotterman, Ph.D.

Georgia State University
Atlanta, Georgia

The history of the development of information systems curricula is reviewed. The DPMA Model Curriculum for Computer Information Systems, the ACM Information Systems Curriculum Recommendations for the 80's, and the IFIP International Curriculum for Information Systems Designers are the current curricula in information systems. These curricula are examined and compared. Appropriate use of standard curricula is discussed.

The history of the development of information systems curricula formally begins with "Recommendations for Academic Programs in Computer Science" (Curriculum 68).[1] It actually begins before that, for many of us were struggling to define "applied computer science" or data processing in the mid- to early-sixties. Some of us had reached academia via the field of data processing, and some of us were housed in business schools. All of us were puzzling over the problem of how best to educate those students whose task would be to develop and implement computer-based information systems in organizations.

Curriculum 68 contradicted our experience. It was hard to see how these courses were going to prepare someone to do those jobs we had just left. Nevertheless, Curriculum 68 was the only standard curriculum available, and it had an impact. Dig deeply into some information systems programs today and remnants of Curriculum 68 can still be found.

In 1972 and 1973, curricula were published which directly addressed the needs of information systems programs.[2,3] In 1974 IFIP published a curriculum aimed at the system designer.[4] Also in 1974, ACM began to reconsider their curriculum in computer science. In 1979 ACM published "Curriculum 78: Recommendations for the Undergraduate Program in Computer Science."[5] This report, under the heading of "Other Requirements and Electives," stated that

> A large portion of the job market involves work in business-oriented computer fields. As a result, in those cases where there is a business school or related department, it would be most appropriate to take courses in which one would learn the technology and techniques appropriate to this field. For those students choosing this path, business courses develop the background necessary to function in the business environment.

Thus we have the computer scientist's vision of information systems.

In 1981, DPMA published the <u>DPMA Model Curriculum for Undergraduate Computer Information Systems Education</u>[6] which is one of the major subjects of analysis in this paper. ACM published "Recommendations for "Master's Level Programs in Computer Science,"[7] "Recommendations and Guidelines for an Associate Level Degree Program in Computer Programming,"[8] and the results of a survey of existing information systems programs entitled "Educational Programs in Information Systems."[9] The survey reported in the latter found 91 schools offering some form of information systems study with 70 programs at the bachelor's level and 54 programs at the master's level. Of 87 programs meeting a minimum set of criteria to qualify as information systems programs, 77% were located in a Business or Management College. Finally, of the names associated with these programs, 27 were Management Information Systems, 18 were Information Systems, 5 Business Information Systems, and a total of 56 of the names included the words <u>information systems</u> in some arrangement. I mention these results because they give some substance to what otherwise could represent merely wishful thinking on our part. A significant number of schools do see "information systems" as a distinct discipline.

In 1982 the IEEE Computer Society developed "Curriculum Recommendations for Software Engineering"[10] but this report has not yet been approved by the responsible body of IEEE and thus cannot be discussed in this paper. Also in 1982, ACM published "Information Systems Curriculum Recommendations for the 80's: Undergraduate and Graduate Programs"[11] which, as in the case of the DPMA report, is one of the major subjects of analysis in this paper. The curriculum activities described above are summarized in Figure 1. Underlined dates in Figure 1 indicate activities specifically directed to information systems.

CURRICULUM ACTIVITIES

1968 CURRICULUM 68 (ACM)

1972 CURRICULUM RECOMMENDATIONS FOR GRADUATE PROFESSIONALPROGRAMS IN INFORMATION SYSTEMS (ACM)

1973 CURRICULUM RECOMMENDATIONS FOR UNDERGRADUATE PROGRAMS IN INFORMATION SYSTEMS (ACM)

1974 AN INTERNATIONAL CURRICULUM FOR INFORMATION SUSTEMS DESIGNERS (IFIP)

1979 CURRICULUM 78 (ACM)

1981 DPMA MODEL CURRICULUM FOR UNDERGRADUATE COMPUTER INFORMATION SYSTEMS EDUCATION (DPMA)

1981 ACM MASTERS CURRICULUM IN COMPUTER SCIENCE (ACM)

1981 RECOMMENDATIONS AND GUIDELINES FOR AN ASSOCIATE LEVEL DEGREE PROGRAM IN COMPUTER PROGRAMMING (ACM)

1981 EDUCATIONAL PROGRAMS IN INFORMATION SYSTEMS (ACM SURVEY)

1982 CURRICULUM RECOMMENDATIONS FOR SOFTWARE ENGINEERING (IEEE)

1982 INFORMATION SYSTEMS CURRICULUM RECOMMENDATIONS FOR THE 80'S: UNDERGRADUATE AND GRADUATE PROGRAMS (ACM)

FIGURE 1

CURRENT CURRICULA IN INFORMATION SYSTEMS

DPMA Model Curriculum for Computer Information Systems (Prepared by the DPMA Education Foundation Committee on Curriculum Development

Figure 2 provides a visual summary of the DPMA Model Curriculum. The apparent three-dimensional figures represent the required courses. The flat rectangles indicate recommended electives, while the circles indicate necessary business support courses. The connecting lines indicate course progression and prerequisite structure. CIS-1 through CIS-4 are lower division while the remainder of the CIS courses are upper division.

The seven required core courses are:

CIS-1 Introduction to Computer-Based Systems
This course is what is sometimes called the "computer literacy" course. 50% of the course is devoted to introductory hardware, software, processing and data communications concepts, 10% to the future of computers in society.

CIS-2 Applications Program Development I
An introductory programming course using COBOL. Emphasis throughout the course is on business applications and programming techniques relevant to those applications.

CIS-3 Applications Program Development II
An advanced programming course with 35% of the course devoted to design concepts. The language is still COBOL.

CIS-4 Systems Analysis Methods
The first course in systems development. Emphasis in the course (60%) is on documentation tools and techniques with the remainder of the course dealing with an overview of the systems development life cycle, derivation of the logical system and information gathering and reporting.

CIS-5 Structured Systems Analysis and Design
An advanced course in systems development with emphasis on structured techniques. 25% of the course deals with documenting the current physical system and derivation of the current logical system, and 60% deals with modeling the new logical system, derivation of the new physical system, and detailed design.

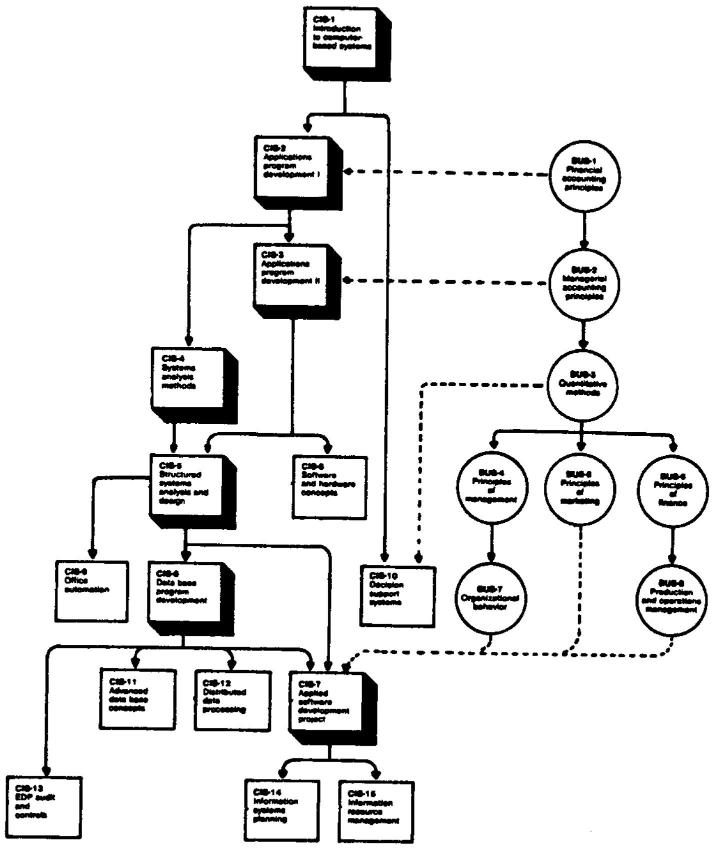
CIS-6 Database Program Development
An introduction to program development in a data base environment. 45% of the course deals with alternative data models, 20% with file organization, 10% with data structure and 10% with data-base administration.

CIS-7 Applied Software Development Project
This is a capstone course which utilizes the team approach to analyze, design and document "realistic systems of moderate complexity." Project management concepts (20%) and communications (15%) such as interviewing and writing skills are included.

In addition to the seven core courses described above, the model curriculum requires that three courses be chosen from the following set of eight recommended elective courses.

CIS-8 Software and Hardware Concepts
A survey of concepts including computer systems components (10%), main storage

STRUCTURE OF DPMA MODEL CURRICULUM FOR COMPUTER INFORMATION SYSTEMS*



* DPMA EDUCATION FOUNDATION, DPMA MODEL CURRICULUM FOR UNDERGRADUATE COMPUTER INFORMATION SYSTEMS EDUCATION, P. 15.

FIGURE 2

organization (10%), instruction sets and data representation (20%), operating systems (20%) and secondary storage (10%). Emphasis is on the relationship of these concepts to applications software.

CIS-9 Office Automation

CIS-10 Decision Support Systems
This course deals with high level information systems which support the management user. The course includes systems and information concepts, systems planning, systems architecture, taxonomy of information systems appropriate for management and a specific consideration of decision support systems (25%).

CIS-11 Advanced Data Base Concepts
A case study based course dealing with data base management systems. Topics include requirements analysis and the design of a data base, data base technology, selection and acquisition of a data base management system and future trends in such systems.

CIS-12 Distributed Data Processing
This course includes coverage of data communications principles (10%), DDP networks (20%), distributed data base structures (10%), and related hardware and software (10%). 20% of this course is devoted to case studies.

CIS-13 EDP Audit and Controls

CIS-14 Information Systems Planning

CIS-15 Information Resource Management
This course includes coverage of information systems management (20%), organization and control (10%), information systems development (20%), and stages of computer information systems growth (10%).

Finally, the model curriculum requires the minimum set of business support courses:

- BUS-1 Financial Accounting Principles
- BUS-2 Managerial Accounting Principles
- BUS-3 Quantitative Methods
- BUS-4 Principles of Management
- BUS-5 Principles of Marketing
- BUS-6 Principles of Finance
- BUS-7 Organizational Behavior
- BUS-8 Production and Operations Management

The report notes that "at schools that are accredited by AACSB, the common body of knowledge in business satisfies these criteria."

Information Systems Curriculum Recommendations for the 80s: Undergraduate and Graduate Programs (A Report of the ACM Curriculum Committee on Information Systems)

Figure 3 provides a visual summary of the ACM curriculum recommendations. The oblong symbol indicates a prerequisite course, the circle indicates a recommended course at the undergraduate or graduate level and the square indicates a recommended course at the graduate level only. The connecting lines indicate course progression and prerequisite structure.

The curriculum is intended to apply to both undergraduate and master's level programs. The undergraduate program omits IS7 Modeling Decision Systems and IS9 Policy. In addition, the courses common to both undergraduate and graduate programs differ in the time spent on each topic and its level of instruction. This paper will deal exclusively with the undergraduate program.

The prerequisite courses in this program are lower division (sophomore year), and the courses satisfying the AACSB Common Body of Knowledge are not specified. The remainder of the courses are intended to be upper division courses. The prerequisite courses are:

P1 Computer Programming
An introductory programming course dealing with algorithm development,

GENERAL STRUCTURE OF INFORMATION SYSTEM CURRICULUM (ACM)

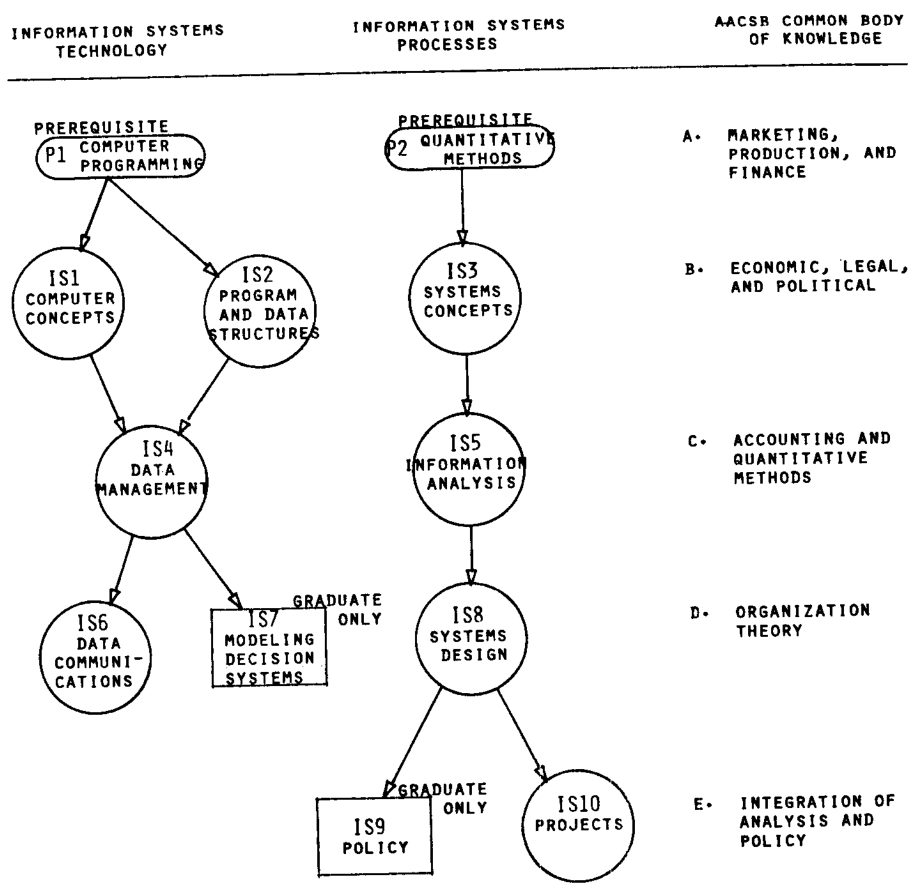


FIGURE 3

programming and computer concepts. Emphasis in the course is on the techniques of algorithm development and programming style. Language specification is "...a high level algorithmic programming language that is widely used." It is intended that this course will have been preceded by a general prerequisite of elementary computer programming.

P2 Quantitative Methods
This course deals with model formulation and application and a study of mathematical programming algorithms and their computer implementations. Problem areas include allocation problems, scheduling, queueing models and inventory models. This course will have been preceded by general prerequisites in finite mathematics and elementary statistics.

The degree program in information systems has three components:

1. IS technology
2. IS concepts and processes
3. Organization functions and management.

Four courses are recommended in the area of information systems technology:

IS 1 Computer Concepts
This course is an introduction to fundamental concepts and terminology of computer architecture, operating systems, and their interrelationships.

IS 2 Program, Data, and File Structures
An advanced programming course using a high level business data processing language (PL/1 or COBOL). Topics include structured programming concepts, data organization and accessing (45%) and design techniques (15%).

IS 4 Database Management Systems
This course deals with the application, logical structure, and physical implementation of database systems. Topics include data structures, operating system topics, database management systems, logical data models, internal data models, database management system facilities, database administration, DBMS evaluation, and distributed databases.

IS 6 Data Communication Systems and Networks
An introduction to the concepts and terminology of data communications, network design and distributed information systems. Topics include communication systems components (25%), networks and control (15%), common carrier services (10%), design of communication networks (10%), and network management and distributed environment (20%).

Four courses are recommended in the area of information systems concepts and processes.

IS 3 Information Systems in Organizations
This course introduces fundamental concepts of systems, information, and information systems. More important, it is a foundation course in that it establishes the role of information systems in organizations. Topics include information systems and organizations (30%), representation and analysis of system structure (20%), systems, information and decision theory (10%), and information systems applications (35%).

IS 5 Information Analysis
The first course in systems analysis and design. The course is directed to information analysis and the logical specification of the system and includes application and development strategies, application system development life cycle, application systems development management, individual behavior and group dynamics in the development process, problem need identification and feasibility assessment, information requirements determination, and requirements analysis and logical specification (30%.)

IS 8 Systems Design Process
The second course in systems analysis and design. Topics include quality assurance review of logical design, the application software make or buy decision, planning to accommodate change, detailed logical design (25%), physical design (25%), hardware and systems software selection, and program development and testing.

IS 10 Information Systems Projects
This is a capstone course. The course uses projects to draw together the concepts of the preceding information system development courses.

In the area of organization functions and management, the report simply recommends that an information systems program satisfy the accreditation standards of the American Assembly of Collegiate Schools of Business relative to the coverage of the common body of knowledge.

The AACSB accreditation standards specify that degree programs in business and administration include in their course of instruction the equivalent of at least one year of work comprising the following areas:

a) a background of the concepts, processes, and institutions in marketing and distribution, production, and financing functions of business enterprise;

b) a background of the economic and legal environment of business enterprise along with consideration of the social and political influences on business;

c) a basic understanding of the concepts and methods of accounting, quantitative methods, and information systems;

d) a study of organization theory, interpersonal relationships, control and motivation systems, and communications;

e) a study of administration processes under conditions of uncertainty including integrating analysis and policy determination at the overall management level.

This is the "common body of knowledge." A study of international business has since been added to this list.

An International Curriculum for Information Systems Designers (A Report of the International Federation of Information Processing -IFIP Technical Committee for Education - TC 3)

Figure 4 is a visual summary of the IFIP curriculum for systems designers. The curriculum is not divided into courses but rather into four modules with the contact hours to be spent in each module specified. The modules are:

Module 1:	Computers and Information Processing Systems--200 hours
Module 2:	Techniques of Management Science--120 hours
Module 3:	Organization Theory and Practice--80 hours
Module 4:	Information Systems Desigr --300 hours

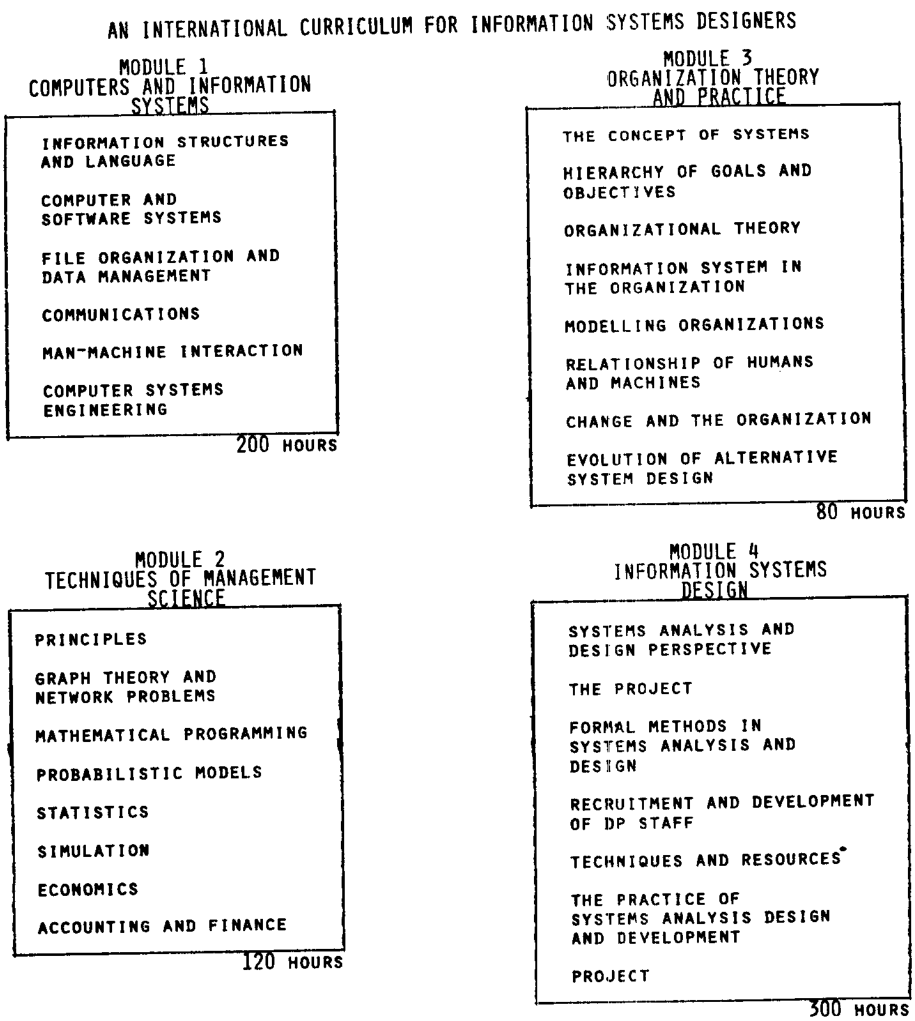


FIGURE 4

The report claims that the curriculum has been constructed so that it can be covered within twelve months of formal study including a substantial amount of practical work. The report also states that the formal coursework should be supplemented by at least 6 months of project work "in the field."

<u>Recommendations and Guidelines for an Associate Level Degree Program in Computer Programming</u>
(A Report of the ACM Committee on Curriculum for Community and Junior College Education)

This, of course, is not an information systems curriculum and its authors did not intend that it be considered as such. The targeted occupation (entry-level applications computer programmer), however, is within the field of information systems. I have included the lower division but complete program with those designed for four year programs.

This curriculum is presented in terms of topics and no attempt is made to specify courses or even to specify relative weighting. The report presents an additional finer level of detail than the one presented in Figure 5.

In addition to the topics shown in Figure 5 the report notes that

> The computer programmer working in the computer applications area should be acquainted with certain fundamentals of the business and economic systems. . . . Knowledge of the functions of a commercial organization and a comfortable use of its vocabulary are important. Skills needed in this area are often met through introductory

AN ASSOCIATE LEVEL DEGREE PROGRAM IN COMPUTER PROGRAMMING

TOPICS IN PRINCIPLES AND TECHNIQUES OF PROGRAMMING	PROGRAMMER ENVIRONMENT TOPICS
• DATA REPRESENTATION, STRUCTURE, STORAGE, AND PROCESSING	• COMPUTER EQUIPMENT AND FUNCTION
• PROGRAMMING LANGUAGES AND LOGIC	• PROGRAMMING LANGUAGES IN ORGANIZATIONS
• INTERFACE WITH HARDWARE AND SOFTWARE	• COMPUTERS IN ORGANIZATIONS
OPTIONAL ADDITIONAL TOPICS	• OVERVIEW OF AN EXISTING APPLICATIONS SYSTEM
• ADDITIONAL SPECIALIZED PROGRAMMING SKILLS	• OVERVIEW OF THE SYSTEMS CYCLE AND THE PROGRAMMER'S ROLE IN THE PROJECT
• OPERATING SYSTEMS AND JOB CONTROL	
• DATA COMMUNICATIONS AND TELEPROCESSING	• DOCUMENTATION
• DATABASE AND ITS MANAGEMENT	• DATA ELEMENTS AND FILES
• COMPUTER OPERATIONS AND OPERATIONS MANAGEMENT	• REPORT REQUIREMENTS AND FORMS CONTROL
• COMPUTER HARDWARE AND SERVICES	• QUALITY PROGRAMMING WITH STRUCTURED APPROACH
• APPLICATIONS SOFTWARE PACKAGES	
• SYSTEMS ANALYSIS FOR INFORMATION SYSTEMS	• PROGRAMMING PROJECTS CONCEPTS
• SYSTEM DESIGN FOR INFORMATION SYSTEMS	• THE PROGRAMMING PROFESSION

FIGURE 5

business or economics courses. Further knowledge of information flow in an organization can be gained from a course in accounting.

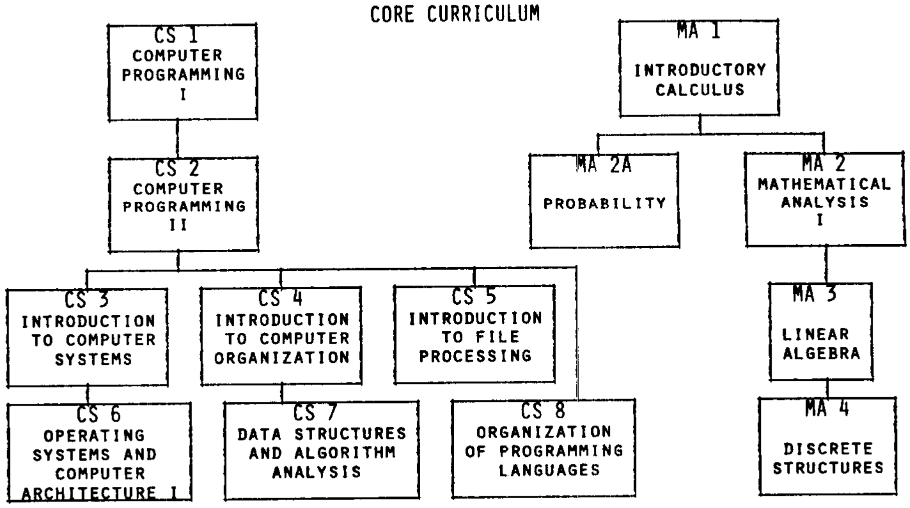
The report emphasizes that the curriculum is designed to prepare students for jobs and not for transfer to a four year institution. On the other hand, the report urges departments with two year programs to articulate their programs with nearby baccalaureate degree granting institutions in order to facilitate transfer.

Curriculum 78: Recommendations for the Undergraduate Program in Computer Science
(A Report of the ACM Curriculum Committee on Computer Science)

This curriculum is included solely for comparison purposes. It is not an information systems program. As a point of curiosity, however, one of my colleagues argues that an optimal information systems program would consist of the implementation of Curriculum 78 in a college of business. During the lengthy process

CURRICULUM 78

CORE CURRICULUM



ELECTIVES

CS 9 COMPUTERS AND SOCIETY

CS 10 OPERATING SYSTEMS AND COMPUTER ARCHITECTURE II

CS 11 DATABASE MANAGEMENT SYSTEMS DESIGN

CS 12 ARTIFICIAL INTELLIGENCE

CS 13 ALGORITHMS

CS 14 SOFTWARE DESIGN AND DEVELOPMENT

MA 5 MATHEMATICAL ANALYSIS II

MA 6 PROBABILITY AND STATISTICS

CS 15 THEORY OF PROGRAMMING LANGUAGES

CS 16 AUTOMATA, COMPATIBILITY, AND FORMAL LANGUAGES

CS 17 NUMERICAL MATHEMATICS --ANALYSIS

CS 18 NUMERICAL MATHEMATICS --LINEAR ALGEBRA

"A LARGE PORTION OF THE JOB MARKET INVOLVES WORK IN BUSINESS ORIENTED COMPUTER FIELDS. AS A RESULT, IN THOSE CASES WHERE THERE IS A BUSINESS SCHOOL OR RELATED DEPARTMENT, IT WOULD BE MOST APPROPRIATE TO TAKE COURSES IN WHICH ONE COULD LEARN THE TECHNOLOGY AND TECHNIQUES APPROPRIATE TO THIS FIELD. FOR THOSE STUDENTS CHOOSING THIS PATH, BUSINESS COURSES DEVELOP THE BACKGROUND NECESSARY TO FUNCTION IN THE BUSINESS ENVIRONMENT."

FIGURE 6

which led to the creation of Curriculum 78, concern was occasionally expressed that the need for educated individuals in "data processing" would not be met by the then developing curriculum. My own feeling was that it was inappropriate for the Curriculum Committee on Computer Science to address the need for information systems education. I felt then that the comittee would inevitably underestimate the unique requirements of information systems education. The paragraph at the bottom indicates the accuracy of that feeling.

A COMPARISON OF CURRICULA

PROGRAM OCCUPATION TARGETS

DPMA	"TO PROVIDE GRADUATES WITH THE KNOWLEDGE, ABILITIES AND ATTITUDES TO FUNCTION EFFECTIVELY AS APPLICATIONS PROGRAMMER/ANALYSTS, AND WITH THE EDUCATIONAL BACKGROUND AND DESIRE FOR LIFELONG PROFESSIONAL DEVELOPMENT."
ACM (INFORMATION SYSTEMS)	"THE GRADUATE OF A PROFESSIONAL I S PROGRAM SHOULD BE EQUIPPED TO FUNCTION IN AN ENTRY LEVEL POSITION AND SHOULD HAVE A BASIS FOR CONTINUED CAREER GROWTH. . . . IN GENERAL THE ENTRY LEVEL POSITIONS ARE: 1. SYSTEMS ANALYST . . . 2. APPLICATION PROGRAMMER OR PROGRAMMER ANALYST . . . 3. INFORMATION SYSTEMS SPECIALIST . . ."
IFIP	". . . THIS CURRICULUM AIMED TO PROVIDE: 1. AN EDUCATION PREPARING FOR A PROFESSIONAL CAREER FOR INFORMATION ANALYSTS AND SYSTEMS DESIGNERS . . . ."
ACM (COMMUNITY COLLEGE)	"THIS PROGRAM IS DESIGNED TO PREPARE COMPUTER PERSONNEL FOR ENTRY-LEVEL JOBS IN APPLICATIONS COMPUTER PROGRAMMING. . . . THE RECOMMENDATIONS GIVEN SHOULD ENSURE SUFFICIENT FOUNDATION SO THAT GRADUATES WITH EXPERIENCE AND CONTINUED LEARNING MAY ADVANCE IN ANY OF A WIDE VARIETY OF CAREER PATHS. . . ."
ACM (CURRICULUM 78)	". . . IT MUST BE RECOGNIZED THAT A PERSON WHO IS GOING INTO THE COMPUTER JOB MARKET AT THE BACHELOR'S LEVEL WILL, IN ALL LIKELIHOOD, INITIALLY BE A SYSTEMS, SCIENTIFIC, ENGINEERING, OR BUSINESS PROGRAMMER. . . ."

FIGURE 7
(UNDERLINES ADDED)

Perhaps not too surprisingly, the occupation target most commonly addressed by the curricula described above is the applications programmer (see Figure 7). The IFIP curriculum is the only one which does not envision programming as a possible career outcome for its graduates. On the other hand, both the ACM community college curriculum and computer science curriculum specify programming exclusively as the starting point for their graduates.

The DPMA, ACM information systems, and IFIP curricula all include some type of analyst position as a potential occupation for graduates of their programs. This is a potential point of distinction of information systems curricula. It is certainly a point of commonality in the three curricula which describe themselves as "information systems."

The 1972 ACM Curriculum Recommendations for Graduate Professional Programs in Information Systems categorized its thirteen courses into four groups.

Course Group A: Analysis of Organizational Systems
Course Group B: Background for Systems Development
Course Group C: Computer and Information Technology
Course Group D: Development of Information Systems

The two courses in Course Group B are:
B1. Operations Analysis and Modeling; and
B2. Human and Organizational Behavior.

Group B is thus closely related to Course Group A. This grouping can be summarized by describing information systems as the study of
ORGANIZATIONS,
COMPUTER AND INFORMATION TECHNOLOGY, and
PROCESSES OF INFORMATION SYSTEMS DEVELOPMENT.

This has become an increasingly common definition of the information systems discipline.

information systems development and 46.5% to organizations.

IFIP	"A SUCCESSFUL CURRICULUM WOULD LEAD TO A VERY THOROUGH UNDERSTANDING OF: A) THE PROCESSES WHICH HAVE TO BE CARRIED THROUGH IN APPRAISING, ANALYSING, DESIGNING AND IMPLEMENTING A COMPUTER-BASED INFORMATION SYSTEM. B) THE TECHNIQUES, TOOLS AND METHODS AVAILABLE TO HELP IN MAKING THESE PROCESSES BOTH EFFICIENT AND EFFECTIVE. C) THE ENVIRONMENT IN WHICH THE COMPUTER-USING SYSTEMS WILL BE EFFECTIVELY EMPLOYED. D) THE PROCESSES WHICH MUST BE CARRIED OUT BY OTHER SPECIALISTS, E.G., COMPUTER PROGRAMMERS, OPERATIONAL RESEARCH SCIENTISTS . . . ."
DPMA	". . . NEEDED COMPETENCIES DICTATE THAT THE PROGRAMMER/ANALYST RECEIVE EDUCATION AND TRAINING IN AT LEAST THREE DIFFERENT AREAS--(1) IN SYSTEMS DEVELOPMENT METHODOLOGIES, WHICH PROVIDE THE FUNDAMENTAL PROBLEM-SOLVING APPROACHES USED IN THE PROFESSION; (2) IN TECHNICAL COMPUTER SKILLS, WHICH PROVIDE THE TOOLS FOR IMPLEMENTING THOSE PROBLEM SOLUTIONS; AND (3) IN BUSINESS THEORY, WHICH PROVIDES AN UNDERSTANDING OF THE CONTEXT WITHIN WHICH THE SYSTEMS ARE IMPLEMENTED."
ACM	"THE NATURE OF THE WORK TO BE PERFORMED BY INFORMATION SYSTEMS GRADUATES THEREFORE ESTABLISHES THREE MAJOR KNOWLEDGE REQUIREMENTS: 1) INFORMATION SYSTEMS TECHNOLOGY 2) INFORMATION SYSTEMS CONCEPTS AND PROCESSES 3) ORGANIZATION, FUNCTIONS, AND MANAGEMENT (INCLUDING INTERPERSONAL AND ORGANIZATIONAL BEHAVIOR)."

FIGURE 8

Statements regarding necessary knowledge and skill in the reports of the three information systems titled curricula indicate virtually complete acceptance of the above definition. Figure 8 presents key statements from those reports.

This agreement is equally clear in the topic recommendations of the three curricula.

In the DPMA curriculum, for example, four of the core courses (CIS-1, CIS-2, CIS-3, and CIS-6) can be considered as directed primarily to computer and information systems technology. Of the eight elective courses, three (CIS-8, CIS-11, and CIS-12) are directed similarly. The total curriculum consists of eighteen courses of which four of the core and a likelihood of 1.125 courses calculated for the electives (CIS-9, CIS-11, CIS-12; 3/8 x 3 = 1.125) provides a likelihood that 5.125 courses (28.5%) of the total curriculum will be devoted to computer and information systems technology. Similarly, 4.5 courses (including CIS-10, CIS-13, CIS-14, CIS-18 of the electives) or 25% of the total curriculum is likely to be devoted to processes of

In order to make a similar calculation with the ACM information systems curriculum we ignore the prerequisites and note that in the "Undergraduate Level IS Curriculum Structure" shown in the report all IS designated courses occur in the Junior and Senior years. Also, the AACSB standards call for the equivalent of at least one year of work. If we assume that the AACSB year also occurs in the junior and senior year then we can calculate that organizations (AACSB) accounts for 50% of the course work, while computer and information systems technology accounts for four of eight IS courses occupying the other year (25%). Thus processes of information systems development also account for 25%.

The IFIP curriculum is somewhat easier to evaluate. Out of the total of 700 hours specified for the entire program, 200 hours or 28.6% may be categorized as computer and information technology, 200 (28.6%) hours (120 hours - techniques of management science and 80 hours - organization theory and practice) as organizations and 300 hours (42.8%) as processes of information systems development.

The percentages described above are shown in Figure 9. The ACM computer science curriculum has been added for purposes of contrast.

Nevertheless, the IFIP curriculum clearly adheres to the three part definition of the information systems discipline.

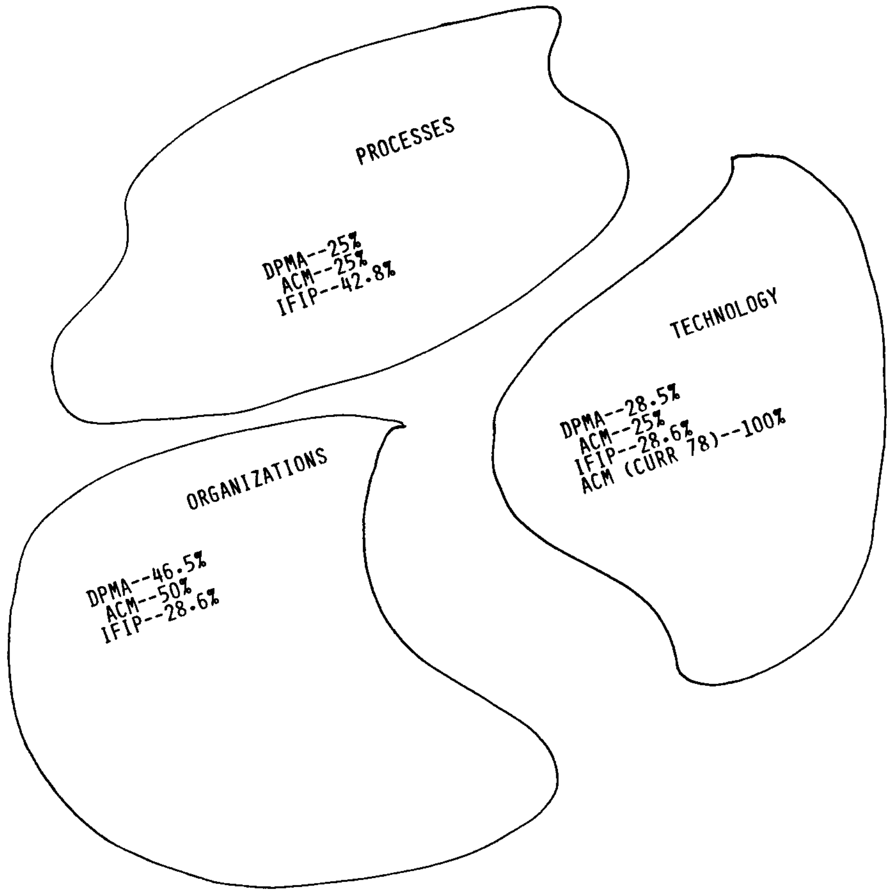


FIGURE 9

Figure 9 makes it clear that there is general agreement between the ACM information systems curriculum and the DPMA model curriculum. The IFIP curriculum devotes about the same relative amount of attention to technology as do the ACM and DPMA curricula but differs in the weighting assigned to the areas of the processes of information systems development and organizations. The IFIP curriculum places approximately 18% more of its total coverage in the area of processes than either the ACM or the DPMA curriculum. On the other hand it devotes approximately 18% less attention to organizations than DPMA and 21% less than ACM. This relative weighting is readily explained by the thrust of the IFIP curriculum. The curriculum is intended to produce information systems designers and one would thus expect a greater emphasis on processes and, as I will indicate in Figure 11 below, the position of systems designers leads one to expect less emphasis on knowledge of organizations.

A direct comparison of the ACM and DPMA curricula indicates that what I have identified as general agreement is specific as well. Figure 10 is a comparison of the ACM and DPMA curricula by subject areas. The diagram follows the basic structure of the ACM curriculum, and courses in the ACM curriculum are indicated by rounded symbols. The dashed circles indicate the two courses (IS7 and IS9) which are graduate only. The squared figures are courses from the DPMA curriculum. Course matching is based on the topic content of the courses as described in the respective reports.

ACM courses P1, IS1, IS2, IS4, P2, IS5, IS8, and IS10 are covered by required DPMA courses CIS2, (CIS1 and CIS8), CIS3, CIS6, BUS3, CIS4, CIS5, and CIS7, respectively. The remaining ACM courses, IS3 and IS6, may be covered by CIS10 and CIS12. The business requirement in the DPMA curriculum and the AACSB requirement in the ACM

A COMPARISON OF THE ACM AND DPMA CURRICULA BY SUBJECT AREA

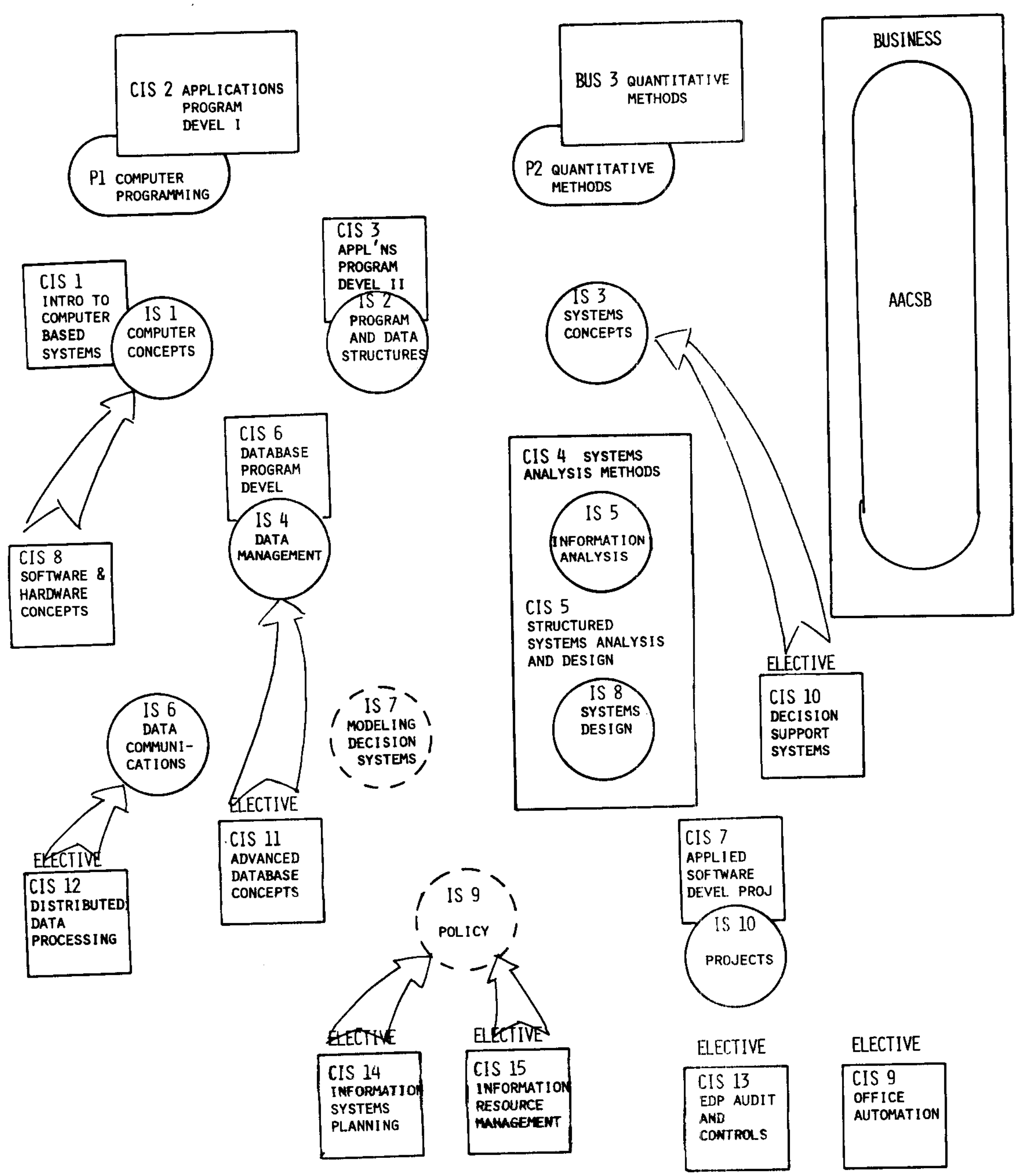


FIGURE 10

curriculum are the same. In terms of topics, the two curricula are very similar and with the selection of two particular electives in the DPMA curriculum become virtually identical. There is a difference between the two curricula in the placement (upper division or lower division) within a four year program. With the exception of the two prerequisites, all of the ACM recommended courses are found in the junior or senior year. In the DPMA curriculum, four of the core courses are placed at the freshman/sophomore level while the three remaining core courses and the three electives are found at the junior/senior level. While it could be argued that the information systems courses in the ACM curriculum are able to assume a more mature student in terms of reasoning ability and both general and business background, I find it hard to see this as an important difference.

THE ROLE OF STANDARD CURRICULA

Before I discuss possible uses of standard curricula I think it is important to register a personal observation on the process of standard curriculum development. I have participated in the development of three standard curricula in one capacity or another, and I admit that this is hardly a scientific sample. Nevertheless, it is my observation that the development of a standard curriculum is typically dominated by a small group of people. This isn't really a criticism, and I have chosen the word dominated carefully. The nature of the volunteer-staffed committees being what it is, a strong driving force is necessary to achieve meaningful results. The driving force is typically a small group of people with shared perceptions and a common set of objectives. Each of the groups I have had experience with has made a determined effort to secure the involvement and contributions of a large and varied group of people who are knowledgeable in the discipline. These efforts have generally been successful, and most standard curricula are able to offer long lists of acknowledgements to those individuals and organizations who have assisted the project. The dynamics of the process are such, however, that the makeup of the larger group of participants changes substantially and constantly throughout the life of the development process. Continuity, and, in fact, initiation, direction, and conclusion, are provided by the small group described earlier. The important point is that there is likely to be a definite perspective, if not a bias, embedded in a given curriculum. This potential bias is even more likely in the detail of the curriculum. The topic content and treatment of an individual course is almost certain to be the work of a small group and may even be the work of an individual.

Again, I don't view this observation as a criticism. I do view it as a caution to the prospective users of standard curricula. As an example, it would, in my opinion, be a mistake to treat DPMA's CIS-4 and CIS-5 or ACM's IS5 and IS8 as the only ways to organize a set of courses dealing with systems analysis and design. A guide--yes; a necessary approach--no!

I believe that the same sort of thinking should apply to the use of the standard curriculum itself. Figure 11 presents a variety of information system occupations on a continuum representing relative requirements for organizational and technical knowledge. Thus, the left extreme of the continuum indicates computer-related occupations which require a great deal of technical knowledge and virtually no organizational knowledge. I have chosen to mark this extreme with the occupation computer scientist. The opposite extreme is marked with the position information architect, intended to mean a position concerned with information needs and the overall information flow within the organization. The position requires a great deal of knowledge of organizations in general and the object organization in particular and a relatively light knowledge of technology. The occupations of systems programmer, systems designer, applications programmer, systems analyst and information analyst have been placed accordingly. In fact, an occupation is not really one point on this continuum but a segment of the continuum, and it is likely that various segments overlap. Thus the occupation described as systems analyst might actually refer to a segment of the continuum which overlaps the point marked applications programmer as well as the point marked information analyst; of course these, in turn, would actually refer to segments rather than points.

In the context of Figure 11, a standard curriculum should mark off a segment of the continuum and act as a guide for the construction

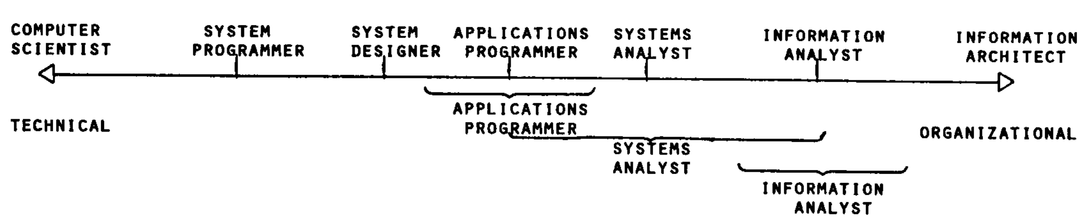


FIGURE 11

of curricula within that segment. The current curricula would seem to offer guidance to segments of the continuum as indicated in Figure 12.

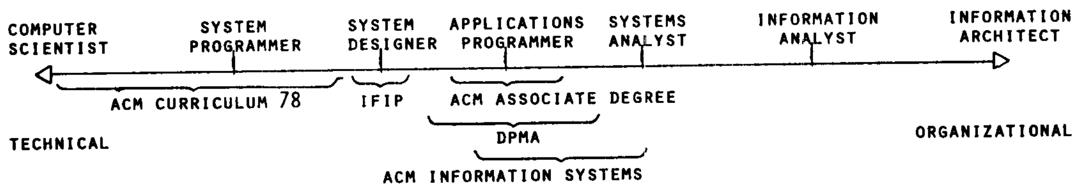


FIGURE 12

An institution considering the development of a program in information systems should first consider its market in terms of the demand for various occupations. Second, what portions of the continuum are served by neighboring educational institutions? Finally, what faculty resources are available or can reasonably be acquired to support a program addressing a portion of the continuum? The first two questions should allow the institution to select a standard curriculum as a guide and further indicate in what direction the curriculum should be modified. The resource question will determine in large measure the specific program that is offered.

CONCLUSION

Figure 13 presents what is perhaps a highly biased view of the supply of and demand for college graduates relative to the continuum discussed above. The supply and demand curves are obviously not correctly proportioned since a correctly scaled demand curve would be much higher than a corresponding supply curve. The invisible y-axis on Figure 13 represents numbers of graduates (positions).

Curve A is the estimated shape of the supply curve of college graduates appropriately trained for positions on the computer-related continuum of occupations. The shape is certainly debatable, but the preponderance of computer science graduates over graduates in information systems is not. The number of information

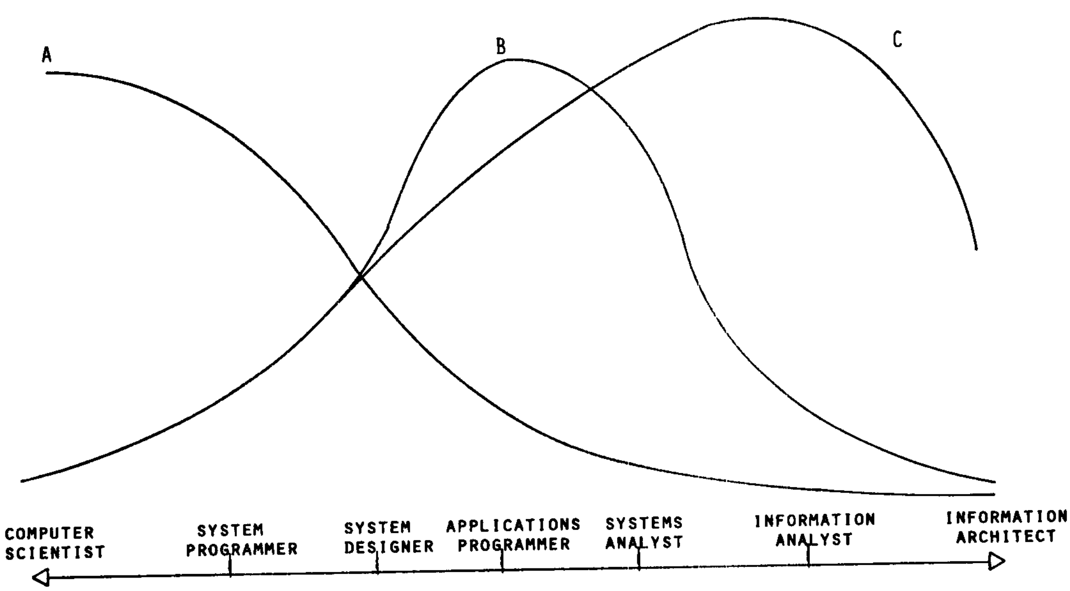


FIGURE 13

systems programs is growing, but they are clearly vastly outnumbered by computer science programs and hence number of graduates. Curve B is an estimate of the shape of the demand curve for college graduates to fill positions on the continuum. Again, the exact shape is arguable, but that demand peaks somewhere over applications programming and systems analysis is supported amply by available statistics. The point is that there is a significant mismatch of supply of and demand for college graduates in computer-related occupations The fault belongs to academia. Computer science says science, and that is a safer flag to fly. We don't want to be accused of crass commercialism--preparing students for the job market. It is almost the ideal situation. We can provide a scientific education while actually preparing students for a field that really does have jobs--lots of them. Never mind that we aren't adequately preparing them for the majority of those jobs that really exist.

The information systems related curricula are late but welcome nevertheless. There is a surprising consensus on information systems education. We agree on the target occupations, we agree on the fundamental knowledge areas, and we even agree on many of the courses and topics which should be taught in an information systems curriculm. Perhaps this consensus will give courage to our academic brethren. Perhaps we can begin some badly needed information systems program and answer the demands of industry.

Lest I leave you with too positive a view, however, regard Curve C. There is now some evidence that demand is continuing to shift, that the locus of need is moving into the user organization. The peak demand may be moving toward the information analyst--that person in the user organization who is technically competent yet intimately familiar with the organization and knowledgeable in organizations in general. Curve C may thus be the demand curve that we are now or shortly will be facing. Now regard the gap to the right of the DPMA and ACM curricula brackets.

REFERENCES

1. ACM Curriculum Committee on Computer Science. Curriculum 68: Recommendations for Academic Programs in Computer Science. Communications of the ACM 11, 3 (March, 1968), 151-197.

2. Ashenhurst, R. (Ed.) Curriculum Recommendations for Graduate Professional Programs in Information Systems. Communications of the ACM 15, 5 (May, 1972), 364-398.

3. Couger, J. Daniel (Ed.) Curriculum Recommendations for Undergraduate Programs in Information Systems. Communications of the ACM 16, 12 (December, 1973), 727-749.

4. Brittan, J. N. G. (Ed.) An International Curriculum for Information Systems Designers, IBIIC Intergovernmental Bureau for Informatics, Rome, 1974.

5. Austing, Richard H., Bruce H. Barnes, Della T. Bonnette, Gerald L. Engel and Gordon Stokes (Eds.) Curriculum 78: Recomendations for the Undergraduate Program in Computer Science. Communications of the ACM 22, 3 (March, 1979), 147-166.

6. Adams, David R. and Thomas H. Athey (Eds.) DPMA Model Curriculum for Undergraduate Computer Information Systems Education. Park Ridge, Illinois, 1981.

7. Magel, Kenneth I., Richard H. Austing, Alfs Berztiss, Gerald L. Engel, John W. Hamblen, A. A. J. Hoffmann and Robert Mathis (Eds.) Recommendations for Master's Level Programs in Computer Science. Communications of the ACM 24, 3 (March 1981), 115-123.

8. Little, Joyce Currie, Marjorie Leeson, Harice Seeds, John Maniotes, Richard H. Austing and Gerald L. Engel (Eds.) Recommendations and Guidelines for an Associate Level Degree Program in Computer Programming. ACM, New York, 1981.

9. Nunamaker, Jay F., Jr. (Ed.) Educational Programs in Information Systems. Communications of the ACM 24, 3 (March, 1981), 124-133.

10. IEEE Computer Society. Curriculum Recommendations for Software Engineering. IEEE Computer Society, Los Alemedos, California, 1982.

11. Nunamaker, Jay F., Jr., J. Daniel Couger, and Gordon B. Davis (Eds.) Information Systems Curriculum Recomendations for the 80's: Undergraduate and Graduate Programs, Communications of the ACM 25, 11 (November, 1982), 781-805.

IMPROVING THE PROCESS OF TECHNOLOGY TRANSFER BETWEEN THE UNIVERSITY AND INDUSTRY

Edgar H. Sibley

Alpha Omega Group, Inc.
Silver Spring, MD, 20910

Is there a lack of transfer of Information Systems Technology? Under what conditions does this become a problem? Many educators, managers and implementers of large scale information systems say that there is a problem and that it is the most important factor in our future. But is technology transfer a problem only to the Information Systems Profession? And who is to blame? This paper addresses three very important questions:

What is being done to ameliorate the situation?

Are the attempts effective?

How can the professional societies help?

INTRODUCTION

The transfer of technology between any two people or groups of people has probably always been a problem. One can imagine the cave-dweller saying "That will never catch on!" when first being introduced to the concept of the wheel. There are many problems that the fledgeling information systems industry has experienced in acceptance of a new method, enough to make many information systems' professionals assume that the derogatory use of the abbreviation/acronym NIH (Not Invented Here) was indeed invented for the sole use of the information system/data processing community.

In fact, the same or very similar problems of technology transfer apparently exist in many other disciplines: possibly the information industry is doing slightly better than the average. And until some work is done to substantiate or disprove this, it would be wrong to subject the industry to unnecessary flagellation, but instead concentrate on ways to increase the transfer in spite of a perceived misuse of knowledge.

The following is a brief historical extract; it discusses some personal experiences in engineering. It is included because it leads to some conclusions that appear to be valid also for the information systems field.

Technology Transfer in a Different Industry

In the late sixties, I was asked to consult as a mechanical engineering design and computer applications specialist for the R&D division of a major aircraft manufacturer. Our intent was to generate interest in a full scale funded effort by a major governmental agency. The overall philosophy was to discover, by observation and interviews, how a complex engineering structure (in this case an aircraft) was designed, what information was used by the designers in pursuing their task, and how the "raw" engineering graduate was taught the "tricks of the trade." We should then, we hoped, be able to use this material in the specification and ultimate implementation of a "designers' workstation" that could present the latest information to the design team at the right time, as well as help the younger or less experienced to become more effective more rapidly.

Although not entirely successful, we did discover some facts about the use and misuse of information and the problems of its transfer. The results that are significant for our purposes here are:

1. Many of the first round designers were brilliant innovators and artists and not stereotypical engineers. They were most interested in tools that would help them test out high level ideas; they saw little chance of a CAD system "participating" in their "innovative" process. In fact, they found it difficult to describe how they "knew" the best design, or what information they used.

2. At the second level, the designers were detailing the structures. Computerized information system could really help here, because the work was often drudgery. However, the people feared automation, even though they knew about the shortage of engineers. Moreover, they felt that the current CAD system had not really reduced the cost, though it had reduced the delays and maybe produced a better aircraft design.

3. Older designers felt that new graduates were "wet behind the ears" and needed to have their "errors," as learned in their University courses, "straightened out." One example was given by a research engineer: welding had been tried in the fifties and proved unsuccessful (due to welds cracking); thus only "conventional" rivetting and similar fastener technology were being considered. A fresh graduate who asked "Why not weld some of these parts?" was told (in no uncertain terms) that

this had been tried and was not acceptable. The attitude of the "experienced" designers continued long after the "facts" were incorrect due to advances in technology. One of the hopes of the research engineer was that the project might help reeducate designers (many not far into their thirties) who used obsolete technology, but were in control!

It is interesting to note that even in the sixties, the "half-life" of technologists was a serious worry. Engineers were thought to have a five to ten year half-life, depending on their discipline -- with electronics and computing as the shortest, assuming no attempt to keep up-to-date by participation in seminars, attendance at professional meetings, and reading new material (books, technical publications, and trade magazines).

It is left as an exercise to the reader to see how many parallels there are between the "engineering case" and the problems of information system technology transfer now discussed.

WHAT IS THE PROBLEM?

Many information systems projects have failed in spite of the availability of a good and tried technology that could have saved them; too late the management have found that a viable technology was developed elsewhere but was unknown to the designers or implementers, or that a method was known but ignored (for no good reason), or those who knew of it were not consulted or considered "dangerous." Such a situation is hardly "healthy."

Meanwhile, the world-wide demand for information systems designers and implementers [1,2] is still rising. Indeed, the demand substantially exceeds the supply, and this is likely to continue into the next century. Some straight-line predictions (in one sense laughable) are worrisome: they suggest that almost the entire population will be employed in the information industries early in the next century. The following are two examples of runaway needs:

The software demands of the manned space flight program grew from about one million instructions for Project Mercury (the early sixties) to about forty million lines for the space shuttle today.

DoD expenditure on embedded computer software is five to six billion dollars per year today (with a little less than two billion in hardware) [3] but both hardware and software expenditures are expected to increase five to six times by 1990.

Naturally, the larger employers of computing professionals, many also contractors for computer and data processing services, have taken note of these alarming predictions. A major reason for the VHSIC (Very High Speed Integrated Circuits) project [4] and the newly announced "DoD Software Initiative" [5,6] is a realization of the problem of finding ways to develop the necessary information systems more efficiently (i.e., to be less costly and require less of the scarce "people-resource").

The conclusion is obvious: there will not be enough information systems professionals to go round unless something quite drastic is done! But what? People must use the latest and most cost effective technology to amplify the ability of the scarce personnel resource.

IS THIS INFORMATION INDUSTRY SPECIFIC?

The "engineering case" was intended to show that the information industry is not alone in finding it difficult to transfer technology. However, the data and information processing profession does have special problems.

The production of large scale information systems is an area of low productivity with poor performance, and this bad situation is not improving. Perhaps rapid obsolescence of ideas has not affected software production in the past; hence the relatively slow realization that major problems existed. But soon the more obsolete designers and implementers will be further behind as software and hardware interfaces become more "fuzzy." This will be accelerated as results from the VHSIC project, which has been particularly successful, are introduced into complex and large scale systems; the new hardware and software integration needs then may obsolete all but the most forward thinking. And even then, there are many technical prophets who feel that the USA is underexpending in aspects of computer architecture R&D. But the particular problem seems to be the software field, where the greater part of the information systems costs and efforts occur.

It is relatively easy to compile lists of information systems problem areas and their probable cause. Some examples are:

a. Development times are too short.

b. Development is hasty.

c. Well tested development tools are lacking.

d. Program development is poorly structured.

e. Poor documentation makes maintenance difficult.

f. Code is rarely reusable.

g. Interfaces to system tools are not "friendly."

There are undoubted management problems; some of those summarized in a 1981 workshop [7] on software productivity were:

1. Lack of management incentives, measures,etc.

2. Little understanding of software: inflexibility, inexperience in management, etc.

3. Non-people-related costs: lack of tools, etc.

4. Fostering new ideas and creativity; incentives.

5. Managers chosen for technical and not managerial

skills.

6. Difficulty in motivating and retaining high productivity people.

7. Attracting people by non-technical or financial (environmental) means.

The GAO,[8] a governmental watchdog organization, saw a number of reasons for major problems in large scale governmental information systems procurement. These were :

1. Lack of proper guidance on software contracting.

2. Top management failure to commit appropriate resources and adequately trained project officers: they do not realise the importance of and difficulty in software contracting.

3. Committment to contracts for software development before proper definition of the problem.

4. Lack of good management practice in contract monitoring: unexpected effects of scope changes, poor inspection of results, lack of good interface to contractor (an adversary relationship).

5. Lack of control makes it impossible to blame the contractor even when the delivered system does not work.

As a part of this work, GAO surveyed nine systems that cost a total of $6.8M; $3.2M was for software delivered but never successfully used, $1.95M for software paid for but never delivered, $1.3M for software extensively reworked or abandoned, $198,000 for software used after changes, and only $119,000 for software used as delivered.

WHAT IS BEING DONE TO AMELIORATE THE SITUATION?

The problems of information systems technology have been addressed more and more in literature and at conferences in recent years. Of course, the more progressive organizations, the better universities, the agressive seminar companies, and the technical press have been addressing this also ... but there is a large body of programmers, coders, and even system designers who have never joined a professional society, and not read a technical book or article since their university or trade courses; they apparently feel that they know all that is necessary for them to survive until retirement.

Maybe little can be done for these on-the-job unemployed. The "Mongolian Horde" were hired because of the expanding need for large information systems, this partly explains some of the serious GAO comments. But some organizations are seriously trying to raise the overall information systems knowledge level of the staff; unfortunately, however, the real help appears to be focussed on the higher level technologists — an attempt to help them remain current in the major skills (or retrofit as needed).

Some collaborative private sector efforts are first briefly examined for their apparent intent and effect. Then the rather large new software initiative of the DoD is summarized.

University and Industry: Colaborative Efforts

There are many joint industry/university efforts now under way. These generally attempt to transfer technology from a particular University effort to an Industrial environment. Unfortunately, either these efforts are relatively new, or they have not been "tested" to determine in what ways they are successful. Equally, there are problems, as seen by some academics and administrators, in justifying their participation in these programs.

As the President of the National Academy of Sciences, Frank Press, has said[9]: "How does a researcher justify withholding publication for six months for commercial reasons but not when the Government asks him to do the same thing for reasons of national security? How do we deal with the problem of 'eating the seed corn,' luring away the best students and teachers with better corporate salaries? How do we deal with the corrosive, if not destructive, tensions created ... (by) some professors becoming rich?"

This quotation is from a conference on "Partners in the Research Enterprise" held Dec 1982 to bring together leaders in Industry and Academia in electronics, chemicals, and pharmaceuticals. Other speakers at that conference also made interesting points on the issue of transfer of technology.

Robert Nicholas, chief counsel and staff director of the Subcommittee on Investigations of the House Committee on Science and Technology (substituting for Representative Gore [D-Tenn] but expressing Rep. Gore's ideas) felt that the ties could deprive the Congress of unbiassed advice on new technologies; he was also worried about the effect of high research monies on the nature and structure of university research; e.g., in competition for the best graduate students. Others asked if this would further erode the position of the Humanities and possibly affect the popularity of some research areas. Dr. George Low of RPI suggested that the use of real life examples made it necessary to "do some work that is not communicated freely" and "the only real measure of the quality of technology is the marketplace ... the quality of some university work is best judged by its commercial value and hence must be proprietary."

This is not to say that the older programs have not given a great deal of valuable methodology to their sponsors (by some measure), but that the efforts should be evaluated to see which structural portions of industry really benefit and how this might affect university efforts. If, as previously suggested, it is only the high level designers or already knowledgeable computing technologists who benefit, then the "hard-core" (undereducated) may still be in need of a different style of update mechanism.

A few typical efforts are now briefly discussed.

The Massachusetts High Technology Council. The MHTC is asking member companies to give 2% of their R&D budget for higher education support. This includes: endowed professorships, career development professorships, graduate research fellowships, consulting contracts with faculty, student loans, cooperative education programs summer internships, and matching gift programs. From the description, this seems to be primarily a method of increasing the flow of qualified personnel. The MHTC sees a lack of such personnel as a failure of the educational system, and feels that the "rapid rate of change characterizing high technology makes it very difficult for the educational system to keep its capacity current and well tuned to the needs of the market."

University of California's MICRO Program. This cooperative venture between UC and several California companies to further R&D in microelectronics and computer science. The work is carried out by university staff and is financed in equal measure by the company and the state. The industrial contribution may involve reviewing ongoing work and providing real-world problem situations for the researchers. As one of the sponsors has said: "While this relationship doesn't guarantee technology transfer, it at least encourages it. The program is new enough (begun in 1982) that its effectiveness as such a vehicle cannot yet be assessed."

University of Minnesota's MISRC. The Management Information Systems Research Center has been in existence since 1968. The major goals of MISRC are to facilitate research in the field of MIS and to promote a productive research relationship between the MIS practitioner and academia. This may be summarised as research, interaction and education. The firm which contributes to MISRC appoints an Associate Representative, who is the liason between the sponsor and the program. Faculty and the Representatives decide on strategic topics for each year. Activities, other than research, are: programs and workshops using national speakers; discussion groups on issues; and student internships.

The Wang Institute's Corporate Associate Program. This is "designed to strengthen the relationship between companies involved in software development and the School of Information Technology faculty and students." Benefits cited are: attendance at student project presentations and interviews with them (the only recruiting allowed on-campus); faculty colloquia; faculty consultation; special two day seminars; discounts on summer courses; technical reports, and desks for MSE students sponsored by an Associate. The program is less than a year old.

The DoD Software Initiative.

Growing difficulties in producing very large systems today and the fact that future needs for software engineers were likely to exhaust any probable supply has resulted in the announcement of a Software Initiative sponsored by the DoD. Its first public workshop was convened in Raleigh, North Carolina, February 7-9, 1983 with a group of researchers, technologists, and systems managers in both the public and private sector as attendees. The interest generated in the announced aims and levels of funding led to a large response; selection was based on the participants' stated ability to make a contribution to the workshop, with a limit of 400.

The object of the workshop was to "provide a forum for discussing technical issues related to the plan." The overall plan calls for a significant thrust involving added funding of software R&D of a little more than $200M from 1983 to 1989. The intent is to use federal 1983 as the "preparation" phase with about $6M of seed money to plan, arrange other workshops in specific areas (Computer Systems Architecture, Reliability, Management, and Human Engineering), issue preliminary RFP, and investigate a Software Engineering Institute. The next two years involve a "Consolidation" phase, where the funding is rising to almost $57M per year by 1986, followed by an "Enhancements" phase at almost $60M per year (1987-1989). It is then assumed that there will be a "Transition" phase where almost all the effort is passing to the services (1989 on) except for a small continued effort (less than $10M). However, the total R&D Science and Technology Support budget is expected to remain at $100M. All the above budgets are based on 1984 constant dollars (without inflation).

The Institute has some interesting goals, they are intended to support the "effective application of technology to DoD software problems by assimilating software technology advancements into the DoD community's technology base." The specifics include:

1. Identification of Valuable new technology.

2. Evaluation of alternatives.

3. Demonstration of utility.

4. Integration of new technology.

5. Transfer of technology.

6. Support of delivered technology.

7. Research on integration and transfer.

CONCLUSIONS

There are obviously many transfer of technology efforts under way, but some seem to be more oriented towards "survival of the elite" than "upgrading the masses." This is unfortunate; there are many people in industry who feel that being selected to go to a seminar is a "gift" that shows a "job well done" or "effort beyond the cause of duty" and not "a much needed update."

One previous report provides some excellent suggestions on technology transfer:

1. The problem is mainly due to the large number of people in the profession and the relatively late arrival of technology. This requires "skill development" with practice in a Programmer Laboratory, publication of standard designs, and a means to

measure and evaluate the people (presumably in a non-threatening way).

2. The managers as well as the technologists need to be trained. This should include contracting officers. The courses should involve: advanced programmer training; management training; user training, and contractor training.

3. Methods of planning and organizing for technology transfer must be developed. This entails: identification of candidates and their selection; methods of involving organizations and personnel in relevant R&D efforts; studying the methods and planning techniques for technology transfer; preparation and planning for transfer; and maintenance, updating and evaluation of technology transfer.

4. Enforcement techniques, with standards that are known and used by all, tool environments to aid in enforcement, and incentives for contractors for quality.

Too many developers, be they internal or contractor, work on a principle of "get the job done." They receive too little feedback, and seem to be rewarded for "poor quality" by lucrative "maintenance contracts" that would be less costly if the contract had been done well in the first place.

The laboratories are needed to answer such questions as: Why does a technology that apparently was successful in one place fail in another? and Why does one manager seem to motivate the team so much better than another? It is only by being able to answer such questions that the software industry will really be "engineered."

How Can the Professional Societies Help?

The future looks dismal if the various efforts do not come to fruition. However, there are many people who are apparently dedicated to improving technology transfer.

Professional societies have always been interested in the topic, as evidenced by their publications, meetings and conferences. However, the other aspects of the problem are still somewhat unexplored. It is a tragedy when a young mind with good new ideas is "put down" or incorrectly told that the information is incorrect; societies cannot really help to solve this problem if the older persons are not associated with that society (and they unfortunately often feel the cost of a membership is unwarranted), but the professional societies are really more powerful as "speakers and consultants to the policy makers." It is the professional societies who should be making themselves available to upper management and the Congress as "friends of the court."

Thus professional societies can aid by making the possibilities better known to the employers, by lobbying for greater governmental understanding and funding, and by educating the public in the possible problems that can result from a future "software gap."

ACKNOWLEDGEMENTS

The works cited have been a great help in structuring the thoughts of this paper; some of these were recommended by colleagues who also participated in valuable discussion. They include: Dr. Jim Ruby of Hughes Aircraft, Herb Willman of Raytheon, Tricia Oberndorf of NOSC, Dr. Jack Kramer of AJPO, Herm Fischer of Litton Data Systems, Dr. Yaohan Chu and Dr. Ray Yeh of the University of Maryland.

REFERENCES

1. U.S. Department of Labor, Bureau of Statistics: "Occupational Outlook Handbook." Bulletin # 2075, Washington, DC, GPO 1981.

2. U.S. Dept. of Commerce, Bureau of the Census: "Population Estimates and Projections." Series P-25, #870, Washington, DC, GPO 1980.

3. Electronic Industries Association, Government Division: "DoD Digital Data Processing Study -- A Ten-Year Forecast." October 1980.

4. Fischetti, Mark A.: "VLSI/LSI Components." IEEE Spectrum, Jan 1983, Vol 20, No 1, pp 43-47.

5. Yeh, Raymond T., et al: "A Report on DoD's Software Technology Initiative." Tech. Report 1161, Department of Computer Science, University of Maryland, College Park, MD, 20742, 1982, 106 pages.

6. Department of Defence: "Strategy for a DoD Software Initiative." Available from the Data Analysis Center for Software, RADC, Griffiss Air Force Base, NY, 13441, 66 pages + attachments and appendices (approx. 400 pages).

7. Munson, J. B. and R. T. Yeh: "Report by the IEEE Software Engineering Productivity Workshop." Proceedings of IEEE COMPCON '81, 1981, pp 330-359.

8. U.S. General Accounting Office: "Contracting for Computer Software Development -- Serious Problems Require Management Attention to Avoid Wasting Additional Millions." FGMSD-80-4, November 9. 1979.

9. Rubinstein, Ellis: " Industry and Academia: How Entwined Should They Be?" Report on a colloquium in Philadelphia, Dec. 15-16 1982, in the IEEE publication: The Institute, Vol 7, no 2, February 1983.

DATA PROCESSING COOPERATIVE EDUCATION AT DELAWARE TECHNICAL AND COMMUNITY COLLEGE

JACK N. O'DAY

DELAWARE TECHNICAL AND COMMUNITY COLLEGE
GEORGETOWN, DELAWARE

ABSTRACT

Institutions that educate people to enter today's computing profession are concerned with maintaining a curricula consistent with the needs of industry. In addition to establishing adequate course offerings, many two year colleges find the need to affect training beyond the classroom or the college computer room. Such practical experience perfecting the classroom skills can often be accomplished through a cooperative education program.

The thesis of this paper is that most businesses prefer to employ people with proven practical as well as theoretical skills. As educators, it is difficult to teach "experience." A discussion is presented showing how the Data Processing Department at Delaware Technical and Community College has attempted to meet these needs through a Cooperative Education Program.

DELAWARE TECHNICAL AND COMMUNITY COLLEGE BACKGROUND

Delaware is a small state with a population of approximately 600,000. There are three counties: New Castle County in the North, Kent County in central Delaware, and Sussex County in the South.

There are four campuses of Delaware Technical and Community College. The Southern Campus, located in Georgetown, was the first to open in 1967. The Wilmington Campus, located in New Castle County, opened in 1968; the Terry Campus, in 1972 in Kent County; and the Stanton Campus, in 1973 also in New Castle County.

Offering a wide range of Associate Degree, Diploma, and Certificate programs and also many Continuing Education courses, all campuses of the College are fully accredited by the Commission on Higher Education of the Middle States Association of Colleges and Secondary Schools.

Many colleges and universities allow direct and full transfer of credits of several courses from Delaware Technical and Community College including some nearby institutions such as the University of Delaware, Delaware State College, Wilmington College, Wesley College, and Salisbury State College in Maryland.

The University of Delaware provides a University Parallel program for students who wish to take a variety of liberal arts courses at the Georgetown and Wilmington campuses. Many students complete two years in the University Parallel program and transfer to the main campus of the University of Delaware to complete a four year degree.

INDUSTRIAL DESCRIPTION

Of Delaware's three counties, the northernmost county is the most industrialized while the central and the southern counties are a mix of industry and agriculture. Since Delaware has some rather liberal corporation laws, many industries have been drawn to Delaware. Fuel and chemical refineries, manufacturing plants of all types, banking operations, textile plants, and perhaps the nation's largest poultry industry can be found in Delaware. DuPont, International Playtex Corporation, Perdue Poultry, Dresser Industries, the State of Delawre, and many banks from other states such as First National Bank of Maryland are a few examples of larger firms that hire our graduates.

In addition to the larger industries, Delaware has many smaller businesses that hire Del Tech graduates. The computer systems in use in these firms include IBM System 3's, 4330's and 4340's, 34's, 38's and System 370 158's. Most installations use IBM equipment, but there are some using Burroughs and NCR equipment. Recently, many small businesses have purchased mini-computers from one of the many vendors such as Apple, Commodore, Radio Shack, Rexon, Texas Instruments, and IBM.

The most heavily used language is, of course, Cobol. The major employers are requesting programmers with a structured ANSI Cobol background. At Del Tech, RPG II, Fortran, Assembler, and Basic are taught. A student must program a high volume of business reports in all languages. Some of the more difficult programs in Cobol III require 1500-2000 lines of code.

In order to provide experience on hardware typical of that used in our business community, the College has recently installed the following:

- -1 4331 with 1 Meg Core
- -1 3278 Console Display Unit
- -1 3505 Reader
- -2 8809 Tape Drives
- -2 3310 Disks with 2 64 Meg Core Each
- -12 3278 Display Units

WHY IMPLEMENT A COOPERATIVE PROGRAM?

Del Tech has enjoyed the reputation for graduating programmers that get hired by industry. Its employment percentage averages 85 percent or better in the many small to medium-sized data processing installations in the area, shops which do not always provide a salary and a career path that is desirable for some graduates. On the other hand the larger installations, until recently, have continued to demand either a four-year degree and/or two years of programming experience. Consequently, many students have started in the smaller installations, gained experience, and then have been hired away by the larger companies. The College has not been totally satisfied with this arrangement; there has not been an easy solution to the problem.

A little over three years ago, the Data Processing Department made some very crucial decisions as to curriculum, staff requirements, and hardware. One of the decisions was to increase communication with the Advisory Committee and to invite several new key people from industry to participate.

Through a more active dialogue between the department and representatives of industry, the department has been able to make some of the area employers see the fallacy of these old practices in hiring four-year degree people. The larger companies often offer higher salaries and a greater scope for career growth; naturally, they want to hire people who have proven themselves. As a solution to the problem, the department suggested some sort of cooperative program where employers could get a look at prospective employees on the job without any long-term salary commitments. The College agreed to work out a program with one of the larger corporations in the area, International Playtex Incorporated. After the Data Processing Department began to look into cooperative programs, the first reaction was "Why on earth haven't we done this before?" It is the Department's opinion that, for a two year technical institution, cooperative education is the greatest improvement in programs since the adaptation of structured programming techniques!

At Delaware Technical and Community College, a cooperative program is distinguished from an internship on the basis of remuneration. If the student works full-time and is paid wages, the program is considered a cooperative program; otherwise, it is categorized as an internship.

STUDENT BENEFITS

The student benefits greatly from the well-planned on-the-job practical experience, regardless of whether he is ultimately hired at the cooperative employer's site. Area employers who have not yet become involved in the cooperative program have indicated a strong preference for a student with a coop background.

One local personnel director stated that he considered the cooperative experience to be more valuable than a full year's experience on the job at some installations. The student has everything to gain and nothing to lose by participating in the cooperative program. All of the cooperative employers the college now works with are convinced that it is the pipeline for the future supply of programmers. Therefore, they have established reasonable pay scales for the cooperative employee. The fact that the student has the opportunity to work in a progressive data processing organization at a good salary is attractive to students. In one case, when a cooperative student was not hired at the end of the cooperative period, she was ultimately hired by another firm. The key factor in her selection over other applicants was her performance in the cooperative program. She feels that she would not have even been considered for employment without the cooperative experience or her transcript and her resume. In the opinion of all of the past participants in the program, the cooperative program was the most valuable experience of their college career.

One student who went through the interview process at two cooperative employer sites and was not chosen for the program stated that the professional interview process at these two installations prepared him for other later interviews in which he was soon hired.

INDUSTRY BENEFITS

A major obstacle that many two-year colleges have to overcome is the four-year degree requirement that many large companies demand. As was discussed earlier, it is natural for a business to want competent people, and one of the methods of screening has been based on the level of formal education. This method is effective in determining the potential of an employee, but even then, the individual is still an unknown quantity for quite some time on the job.

Businesses are reluctant to hire a student with only a two-year associate degree and no experience without some indication of how the individual will perform in a work environment. In one case, a large industry in the area never used to hire students straight from Del Tech. Ironically, their programming staff was made up of approximately one-fourth of our graduates! These students were hired after acquiring experience at other small local businesses. They did, however, hire an employee with a Bachelor's Degree in Elementary Education and an Associate Degree in Data Processing.

Selling the idea of a cooperative program has been easier because our graduates made up a great portion of their programming staff and because of the outstanding performance of the "school teacher with a Del Tech D.P. Degree."

Cooperative employers say that the ability to interview and select a prospective employee and observe him or her in a rigorous work situation is worth the money spent in salary if only one out of three students is hired. The fact that there is no obligation to hire is attractive to employers.

Recently some competition for coop students has developed among the cooperative employers. Last year, the number of cooperative positions exceeded the number of students available enabling students to have a choice of two or more cooperative opportunities. As a result, two of the cooperative employers plan to be on campus well in advance of the end of the last school quarter this year in order to "recruit" cooperative students. One of these employers is the one who previously never hired a graduate from a two-year college. Obviously, these employers see the advantages of a good cooperative program.

Recently the college has received many inquiries from other area corporations about the program and how they might become a part of it. Soon there may not be enough students to go around - a "problem" the department can live with.

COLLEGE BENEFITS

The College and the Data Processing Department have seen many positive results of the cooperative program. Naturally, a curriculum in a community college thrives as a result of healthy enrollment. The Data Processing Department enjoys the present hardware and staff salary structure because it maintains acceptable enrollment and students get good jobs.

The College, as well as other like institutions, has experienced an enrollment increase over the last two years. Likewise, the enrollment in the Data Processing curriculum has virtually doubled. The availability of a cooperative program has been responsible, in part, for some of this increase. The success of the cooperative program has had a positive influence on the accreditation committee's evaluation.

One business that recently located in the area was so impressed with the cooperative program and the overall Data Processing curriculum that it has requested to be included in the program next year and has offered a scholarship for Data Processing students this year.

The college administration feels that the cooperative program fulfills a two-fold need:

-it provides the close relationship with industry that helps keep course content up-to-date,
-it provides a vehicle for employment beyond the classroom experience.

WHAT KIND OF COOP SHOULD BE IMPLEMENTED?

When the Department embarked upon the journey into cooperative education, many ongoing programs in other curriculums at Delaware Technical and Community College and also at other institutions were investigated. It soon became evident that cooperative programs come in many different varieties. Perhaps the most common arrangement is one where a student alternates between working on the job and attending classes at college. The sessions vary in length with different programs but can be from a semester to half a year, usually during the later half of the student's curriculum. In a four-year college, juniors and seniors might coop while at a two-year college, second year students are eligible.

The rationale appears to be that the student should have enough skills to be somewhat productive while he is learning new skills on the job. In this way, the cooperative employer gets some return on his investment in the program.

At some institutions, cooperative programs are mandatory while others are optional. Students who opt to coop receive a different degree from those who do not. At some schools, a student who coops does not receive his degree until after the completion of the program, often quite some time after a non-coop student has graduated. Another optional cooperative program allows both coop and non-coop students to receive degrees. With this arrangement the coop student receives additional credits for the cooperative training course on his transcripts.

Some curriculums at Del Tech have mandatory internships which give the student practice in the many hands-on skills on the job in situations not possible in the college environment. Interns do not receive pay for the internship period. Nursing students, for example, are required to intern to gain skills on the various instrumentation used in hospitals.

The Data Processing Department discussed the many possibilities of internship versus coop with members of the Advisory Committee. It was concluded that a "model" Data Processing Cooperative Program would:

- be an optional program until such time that sufficient cooperative employers are available to make the program mandatory
- occur after all requirements for the Associate Degree in Data Processing have been met
- be a paid cooperative program, not an internship
- take place in a medium to large state-of-the-art Data Processing installation
- be an intense program to make the student a productive programmer as soon as possible

The job that the cooperative student is being prepared for is one of an entry level Applications Programmer. Graduates from four-year colleges and programmers with experience are competing for these same jobs. It is encouraging that cooperative employers of the area are committed enough to cooperative education that they feel a coop student should be paid equal to that of a newly hired Applications Programmer.

Since the goal of the cooperative employer is to make Applications Programmers productive as soon as possible, it was felt that the student should have the entire two years of Data Processing courses completed to be eligible for the program.

PROGRAM DEFINITION

The Data Processing Cooperative Program at Del Tech runs during the Summer Quarter each year. The cooperative period is twelve weeks of full-time employment. The student has his degree prior to the beginning of the cooperative period and receives twelve credit hours on his transcript in addition to the credits required for graduation.

The student opting for cooperative training may apply for the program early in the Spring Quarter and is interviewed by the prospective cooperative employer(s) following normal entry interview procedures at each job site. The cooperative employers notify students if they have been selected. No record will appear on the transcripts of students who are not selected. So far, all applicants except one have been selected for the program.

COOPERATIVE TRAINING AREAS:

Although the program of training differs slightly at the different work sites, the student will generally spend time working in the following areas:

- Operations
- Technical Services
- Data Base
- Applications Programming

Approximately half of the twelve-week period is spent in Operations, Technical Services, and the Data Base group. The least amount of time is dedicated to Operations because the student is not being prepared for a career in Operations but he should realize the relationship of his activities as a programmer to those of the Operations Department.

The student spends apprxoimately two weeks with the Systems Programming staff or Technical Services group and another two weeks with the Data Base group.

At the end of this period of approximately five to six weeks, he returns to one of the Applications Programming groups, spending the remainder of the coop period writing programs ranging from simple ad hoc programs to some rather long complex production programs later in the coop period.

INDUSTRIAL SITES

The current cooperative employers include:

- International Playtex Incorporated
- State of Delaware Central Data Processing Department
- First National Bank of Maryland

In the first year of cooperative education, there were three students enrolled; of that number, two were hired. Last year, there were seven enrolled and six hired. All have received at least one promotion since being hired. The seventh student cooped at a new site, the State of Delaware Central Data Processing Department, and staff requirements did not warrant an additional programmer at the end of the coop program. Had there been an opening, the student would have definitely been hired, however. An excerpt from a letter accompanying the student's final evaluation from the cooperative training supervisor states:

> "The cooperative program came to a successful conclusion on August 30, 1982. Though employment of ________ was not possible at this time, I believe ________ received exposure to many aspects of the data processing work environment which should prepare her for future employment. ________ demonstrated a strong determination to master every aspect of hands-on training we provided. She was a very conscientious and quite inquisitive individual which, I feel, are strong attributes for a systems programmer.
> I know I speak for all the supervisors and employees that participated in this program by saying that we are supportive of the Cooperative Program if future candidates are as scholastically prepared as ________."

HOW TO OBTAIN COOPERATIVE EMPLOYERS?

There are many factors that influence a business' willingness to engage in a cooperative program with a two year technical college. Some factors are beyond the control of the college, but three important keys to gaining cooperative employers are:

1. An up-to-date Data Processing curriculum producing graduates that get hired.
2. An active advisory committee with representation from the complete cross-section of typical area businesses.
3. A successfully established cooperative program with a business respected in the area.

The success of a cooperative program is directly related to the caliber of the Data Processing curriculum offered by the educational facility. If employers readily hire graduates, then they will more than likely welcome a cooperative program. Those employers with past graduates on the programming staff are prime candidates as cooperative employers. The reputation of the institution has a direct bearing on the willingness of an employer new to the area to enter a cooperative agreement. For example, a large banking operation which recently located in our area came to Del Tech to seek employees for its Data Processing operation. During "head-hunting" at other area businesses, the firm found that Del Tech graduates were employed at nearly every installation. Consequently, this employer asked Del Tech to establish a cooperative program at its installation.

Many two-year educational institutions maintain a close relationship with the business community through some form of advisory committee. Similarly, Delaware Technical and Community College requires that each curriculum maintain a continuing communication with its respective advisory committees.

This alliance has repeatedly resulted in many benefits for the Data Processing Department at Del Tech. Once representatives from the business community learn that a college is eager and able to train students with the skills businesses require, they will more readily hire graduates. All of the present cooperative employers have representatives on the Data Processing Curriculum Advisory Committee of Del Tech, an invaluable means of establishing the critical business contacts that are necessary to implement a cooperative program.

Once the first successful cooperative program is established, it becomes easier to use this as a referral for new programs at other business sites. Thus, it is even more important that the initial programs work well.

SPECIAL CONSIDERATIONS

The mutual agreement between the cooperative employer and the educational facility must be flexible. Both Del Tech and the employer endorse a "working agreement" outlining the responsibilities of the two parties and the student. Although this agreement is not a contract, it is understood at the outset that the program at a business is contingent upon the constantly changing business climate.

The areas of study during the cooperative program are designed to prepare the student for productive programming assignments in a short period of time. At each installation, much reading is required, including extensive use of audio-visual materials. Del Tech has not encountered any employers interested in using the coop as a source of cheap labor. The program guidelines should exclude such a possibility, and any educational institute should avoid such a liaison.

During the coop period, Del Tech has found that it is imperative to maintain constant contact with the student at the work site because the student needs the support from a source outside the work environment. A Cooperative Coordinator, appointed from the Data Processing Department, makes a visitation at each work site once every other week to monitor the progress of each individual in the program. These individual evaluation sessions with the student, his work supervisor, and the industrial cooperative training supervisor are vital for many reasons. The college coordinator can help clear up questions and concerns the student has that he may be reluctant to discuss with his work supervisor. The student needs to feel that he has an advocate. Many minor problems have remained minor and been handled to the mutual satisfaction of both the employer and the student through communication with the coordinator. Both the student and the cooperative employer appreciate the college coordinator. One employer who had previously been dissatisfied with a cooperative program with another college stated that one of the biggest complaints with the other program was that he never saw a representative from the college during the program. Thus, the presence of an active coordinator is perceived by the cooperative employer as a clear indication of the sincerity of the college to maintain the best possible cooperative program.

Grade determination for the Cooperative Training course can be a complicated process. The Data Processing Department at Del Tech determines the final grade based on the following parameters:

The student is given an evaluation of the assigned tasks in each area of study as:
EXCELLENT - GOOD - AVERAGE - FAIR - POOR

The degree of difficulty of the work areas is taken into consideration. For example, a "good" performance of some tasks in the Applications Programming segment would be weighted more heavily than a similar rating in the Operations area. The work supervisors determine the above performance ratings and review them with each student during each phase of the training. The final grade is determined by the college coordinator.

Another factor considered for the final grade is whether the student was hired by the employer at the end of the coop program. Any student who has been hired has earned an "A." Students not hired have often earned "A's" also.

The grade in the course is perhaps most important to the students not hired. Assuming that the student has a satisfactory grade in the program, most prospective employers put a premium on such work experience.

Even though there is a great deal of screening and interviewing of prospective coop students, it is always possible that an employer might, at some point during the coop period, consider it fruitless for a student to continue. There must be a provision to terminate the program without jeopardizing the student's scholastic record. The employer should not be required to invest resource in an individual who is not going to be acceptable. However, an employer is not expected to drop other coop students the minute he finds those students within the group that he wants to hire. No coop student has ever been dismissed during the coop program, but the policy at Del Tech would be to withdraw the student officially from the course rather than administer an unsatisfactory grade.

CONCLUSIONS

If an educational institution does not now provide a cooperative educational program, it may warrant the minimal effort required to investigate the feasibility of doing so. The benefits possible for the student, the institution, and the cooperative employer are unlimited.

A well planned and supervised cooperative program eases the student into a working-learning environment with the least amount of shock. The college is able to upgrade curriculum through close communication with the new employee and the employer. The employer is able to interview and select a prospective employee and place him in a real working environment without a long-term financial commitment. This affords more students a "foot-in-the-door" at more and larger data processing installations.

In the final analysis, everybody wins - the student, the employer, and the educational institution.

Academic Preparation For Tomorrow's MIS Manager

Gustave J. Rath

Northwestern University

This paper covers the process and content of academic courses for preparing MIS managers of tomorrow. It lists types of courses and materials to be taught.

1.0 INTRODUCTION

Academic preparation refers to college and university training which may take place in departments such as computer science, industrial engineering, or management and may be taken by majors or students in other departments. These students may be at the bachelors or masters levels. There are other post-secondary academic sources of preparation not covered in this paper such as proprietary school and community colleges. Two aspects will be considered in this paper: the process and content of the training. Process refers to the method of training and content refers to what is being taught. We can define four levels of proficiency in the use of MIS.

Literacy — A user is considered literate when he or she is able to appreciate the need for MIS, has seen it in operation, understands key concepts and has seen demonstrations and/or played with a keyboard.

Use — A user is considered to be able to use MIS systems when he or she are able to make queries, enter information, and carry out all the major operations with an established data base. This includes accessing the main frame or micro computer. The user should also be able to use documentation for the system being learned.

Development — A user is considered to be able to develop a data base. He or she should be able to take a general purpose MIS package and format the data base. Organize the data Aquisition process. Get the data base up and running. Service the needs of users. Present and sell the MIS system in the organization. Choose appropriate data base language and be able to analyze the cost-benefit of candidate systems.

Programming — A user who can program in assembly language, BASIC, COBOL, FORTRAN, PL/1, PASCAL or other similar languages for the purpose of structuring a data base.

A modern well prepared MIS manager should be able to at least do the first three activities, that is be a user and a developer.

2.0 TYPES OF TRAINING EXPERIENCES

Colleges and universities offer a wide range of educational experiences. In this section I will present a range of experiences based on what I have observed at Northwestern University and looking at other schools. These range from traditional (lectures) to non-traditional (CAI).

2.1 Lecture Courses

Lecture courses are the traditional, cheapest and least effective approaches. The use of handouts, especially when geared to good AV, enhance lecture approaches, but require extensive preparation. The upgrading of academic lectures to the use of high quality aids is usually more expensive than that which schools and faculty are willing to invest. One of the best uses of lectures has been to bring in guests who are MIS managers, producers and designers. This allows the students to see real applications and talk to MIS managers. At times students have experiences which are worth presenting. Students typically read books, write papers and/or take exams whith these courses.

2.2 Programming Courses

Programming courses may have some lecturing, but the focus is on writing codes, and writing programs. Unfortunately, good documentations is usually omitted because of time constraints and lack of importance in the eyes of instructors. The lack of native speaking English assistants and instructors pose additional difficulties in documentation. Courses may use main frames or micro-computers. Students are typically given a series of problems to solve over the period. Novel approaches must be used to test students and grade them.

2.3 Computer Aided Instruction (CAI)

Computer aided instruction (CAI) is usually used as a supplement to courses. At Northwestern we have a program called RIQS which is an interactive data base language. It has a tutorial CAI program which teaches you how to use the data base. Unfortunately, this system has not been supported since the developer of the CAI package ledt Northwestern. The military has used this approach for many years to train users of command and control systems. The preparation of high quality CAI materials may take 50 hours for each hour of instruction. There is a large investment required to develop these materials. The rapidly change state-of-the-art of MIS systems, the high cost of development, and the lack of appreciation for developments of educational materials (vis-s-vis doing research) mitigate against the development of good materials in colleges and universities. Commercial sellers of data base systems who develop such packages could find them being used in colleges and universities.

2.4 Project Courses

Project courses are courses where a project is the major student activity in the course. There may be lectures and presentations, but students are mainly graded on a single project. Other project courses are simply a major set-up, supervised and graded by the instructor. These projects may be done alone or in groups. I have run MIS projects in these types of courses with a range of 1 to around 100 students. Individual data base for children with developmental disabilities. One large project was to develop a data base on MIS system hardware, software, literature and human resources.

2.5 Research and Projects

The most significant opportunities to learn MIS system management comes from independent work. Independent work may be in the form independent study, honors projects, thesis, dissertations, and research. One honors project was the design of a data base for undergraduate academic records in our department. Another involves the design of a data base for drug interaction effects for the VA. A thesis centered on the design of an evaluation data base for the Midwest Region Spinal Chord Injury Care System (MRSCICS). The experience of carrying out a project from systems analysis through implementation is one of the best experiences for the future data base manager.

3.0 LOCATION OF EXPERIENCE

There are three locations in which one may carry out the MIS learning experiences: school, other sites where the student has access (his job, family business, etc) or a research site. The most important criteria for a site is that they really want an MIS system. Other considerations are computer hardware, software, and other technical resources.

3.1 At School

Colleges and universities have a large number of potential customers from alumni funds to property management. Most of them also have a variety of computers and software packages which are access= ible. Arrangements have to be made with specific groups, and there usually is a limit of how many projects may be carried out simultaneously. Convenience is a prime factor in using clients in ones own school.

3.2 Accessible Sites

Students have many potential resources including past employers, current employers, family businesses, business of friends and relatives, coop-placements, local sites (such as libraries, churches, etc.) and coop-placements, local sites (such as libraries, churches, etc.) and places which the faculty have acquired. Very often not-for-profit organizations are especially good as they are in need of such systems and have less problems with proprietary information than business and industry. In some cases students may follow the same site through several courses. At times students have acquired part or full time employment from these ventures.

3.3 Project Sites

When faculty have grants, research contracts or consulting assignments other opportunities open up. At times these are not the best sites for faculty may exert too much control on the project.

4.0 CONTENT OF EXPERIENCE

Both specific courses and general courses are desirable.

4.1 General Courses

Courses in techniques such as systems analysis, operations research, engineering economics, and industrial/organizational psychology are desirable.

4.2 Computer Language Courses

Students should be familiar with main frame and micro computers. They need to have some experience and knowledge of machine and assembly language. They should have some experience with higher level languages and know how to program in at least two of them. BASIC, COBOL, FORTRAN, PASCAL AND PL/1 are candidates. If students Have learned two or more languages it will be easier to learn others in the future.

4.3 Data Base Management And MIS Courses

Course in the theory and structure of data bases (such ideas as relational data bases) will cover the general material needed. Query languages and applications from library science are of interest. Methods of planning, developing, and implementing software and data bases are relevant. Concern for the personnel, management and economic aspects are important. The methods of specifying data bases are important. The issue of documentation needs to be covered thoroughly. How one evaluates data bases needs to be covered.

4.4 MIS Package Training

Depending on the facility the students may have available of data base management languages such as SIR, RIQS, STAIRS, RIMS, etc. Micro computers have available languages like DB Master, Visifile, etc. These languages are usually available for a course, but are not the sole content of a course. Exposure to various data bases is another important feature. Student should be acquainted with such utilities as CompuServe, Lockheed and The Source.

TEACHING SYSTEMS ANALYSIS AND DESIGN IN A COMMUNITY COLLEGE

Mary E. Rasley, CDP
Assistant Professor of Data Processing

Lehigh County Community College
2370 Main Street
Schnecksville, PA 18078

Abstract

The Community College data processing curriculum must serve two masters:

1. Industry

and 2. Four-year transfer institutions

This situation presents a challenge in the area of development of a systems analysis and design course. This course must include parts of CIS-4, CIS-5, and CIS-7 of the DMPA Model Curriculum in order to properly prepare students for industry positions. A detailed description of a course in use at Lehigh County Community College is presented in this paper as a proposed model for the CIS curricula.

Introduction

The Lehigh County Community College (LCCC) Data Processing Curriculum was developed in 1968 to produce entry level programmers skilled in commercial applications. The curriculum originated in and remains a part of the business division of the college. With the increasing number of four-year institutions developing business information systems curricula, LCCC has had to serve two masters: industry and four-year transfer institutions. To meet both needs a course called EDP 203 Systems Development and Design was placed in the curriculum as the capstone course. The course description is as follows:

> Analysis, design, and implementation of data processing systems including design of reports, forms and cards, equipment studies, data collection and data communication systems. Introduction to critical path systems, formal presentation of system evaluation and an implementation schedule are discussed.

This course incorporates parts of the DMPA Model Curriculum courses CIS-4, CIS-5, and CIS-7. The purpose of this paper is to describe the learning objectives of the course and their implementation in a two-year college. I further posit that CIS-4 is insufficient for two-year graduates entering the work environment of corporations today. Two-year graduates entering industry must be equipped with the skills developed in EDP 203 in order to function effectively in today's industry environment.

Course Objectives

The course objectives for EDP 203 are stated as follows:

1. To gain an understanding of the role of the business systems analyst plays in the Management Information Department.
2. To gain an understanding of the process of systems analysis and systems design by course readings and by participating in a systems team project.
3. To gain an understanding of the human relations aspects of the data processing work place.

The structure of the course involves a theoretical base of systems analysis and design and a term project used to reinforce the theoretical framework.

Theoretical Base

The course theory hinges around a model of top-down systems development as described by Marjorie Leeson. Imperative in class lecture topics is the inclusion of the need for the analyst to develop interpersonal skills. This need is given relatively little coverage in most analysis textbooks and, therefore, must be addressed by the instructor. Concepts covered in this framework are: group theory, brainstorming, communication, problem solving, leadership skills, controversy and creativity, power theory, stress and DP careers.

Implementation of Course Theory

Students enrolled in EDP 203 are required to participate in a term project assigned to teams in the class. These teams consist of four individuals ideally, although smaller teams may be used as necessary. Each member of a four person team is required to submit one deliverable. This deliverable encompasses 50% of the students' project grade. The other 50% is derived from a formal presentation of the system delivered by the team to the instructor and system users. The deliverables include:

1. Functional specification
2. User Manual
3. Detailed Systems Documentation (see attachment for detail)

My belief is that a system must be real in order to have meaning to the students. My students concur. To provide this reality I enlist the cooperation of faculty and administrators of the college. These individuals have approached me with automation needs. I spend time discussing the systems with these individuals and if their needs can be met in a 15 week time frame, I offer them a group of students to work with. It must be noted that these faculty and administrators must be prepared for this experience ahead of time. I request that they be as natural, in a business sense, as possible.

I expect the student to deal with user personalities and inconsistent time schedules just as they would in industry. I carefully monitor student-user feedback in the analysis phase to ensure that the solutions they arrive at in the design phase will meet user criteria and will be doable in a semester time frame. I find it vital that I, as the instructor, do not act as a go-between for the students and the users. All problems and questions of a systems nature coming from the students are placed back into their realm of responsibilities. When they reach a stalemate on a given issue I encourage them to brainstorm solutions and as a group evaluate and choose a plan of action. I meet weekly with each group in a lab situation to be updated on their progress and each individual is required to file a formal weekly status report throughout the project duration.

Hardware

To familiarize students with as much a variety of popular hardware as possible, I require that students design their systems around a microcomputer configuration. The school currently owns:

- 8 - Apple II+ computers
- 3 - printers (2 thermal; 1 impact)
- 11 - floppy disks
- 1 - hard disk (20 meg)

This allows the students real-time capabilities to design menu-driven, user-friendly systems.

TEAM PROJECT

Introduction

The description of this project is organized into five parts. They are:

I. GENERAL INFORMATION ABOUT THE PROJECT
II. DELIVERABLES AND DEADLINES
III. THE FORMAL PRESENTATION
IV. GRADING

I. GENERAL INFORMATION ABOUT THE PROJECT
  A. The learning objectives of the team project are twofold:
    1. To gain an understanding of systems analysis and design by actually analyzing and designing a system.
    2. To learn how to work with others on a project team.
  B. The rules of the project are:
    1. The project work will be accomplished in teams.
    2. Each team selects its own members; NO switching of team members may occur unless the instructor switches the teams.
    3. THE ENTIRE TEAM WILL ANALYZE AND DESIGN THE SYSTEM.
    4. Each member of the team is responsible for completion of a:
      a. Specific project deliverable.
      b. Weekly status report (beginning with the first class meeting of the third week reporting work done week #2). The dates are shown on the lecture and lab schedule for the course.
      c. Participation in the formal presentation.
    5. No team may share work or people with any other team.
    6. If the team decides to have one member act as a leader, that member must agree to act as the leader (unanimous decision of all team members). This leader may only be responsible for project coordination, and reports as outlined in #4 above. The leader is not a person to whom unfinished materials are passed.
    7. All personality conflicts and team problems must be worked out by the team. I will only get involved if a team member withdraws from the course or violates course attendance policy.

II. DELIVERABLES AND DEADLINES

Each deliverable is to reflect the decisions made by the user and the entire team. The person responsible for the deliverable seeks agreement from the team on all aspects he/she must include in the document.

  A. 3 Person Teams
    1. Due during lab - 3rd week (2/1/83) - The team contract. Each member of the team is to agree to be responsible for a specific team deliverable. This agreement is signed by each team member. (I will provide the agreement.)
    2. Each week a status report is due from each team member stating the nature of the work and the hours they spent the previous week on their deliverable or their portion of the presentation. The 1st status report is due the 1st class meeting of the third week (reporting 2nd week's work). The last report is due the 16th week reporting the work done the 15th week.
       Format: 1 typewritten page briefly stating
         1) member's name
         2) team #
         3) week ending date
         4) hours spent that week

5) what work was done If no work was done, turn in a page stating that no work was done for #5 above. I will hold all status reports until the day of the final exam when they will be returned.

This accounts for (1 point per week) a total of 13 points of the project grade. No grade is given for the amount of work done each week--so be honest. These reports must be turned in on the night due--if you are absent, mail it to me at the college address on the policy sheet.

3. One team member is responsible for submittal of Functional Specifications on or before 3/15/83.

A functional specification is the "contract" or agreement between the user and the project team. It states in "non-computerese" terms exactly:

a. Why the system is being developed (benefits and costs of the system). These must be quantified.
b. What "the system"* will do for the user.
c. How it will be done by the system.
d. When it will be done.
e. What responsibilities the user of the system has (input preparation, output verification, etc.).
f. The criteria the user will judge acceptance or rejection of the system by.
g. Detailed implementation plan.

No programming or "computerese" terminology may be included.

A suggested format is as follows:

a. Table of Contents
b. General Narrative - A general introduction to the system. Goal and objectives of the system are stated. The benefits to be derived from the system are considered and contrasted against costs of the system. (If exact costs are not known, they may be estimated.)
c. Inputs to the System - All forms and input documents the user is required to submit for system processing and how the data is to be captured (paper, entered by user, via keyboard, etc.). Actual forms or screen formats may be included in the section but instruction on how to fill them out should not be -- those instructions belong in the user's manual. A timetable for submittal of this data, if necessary, must be included. A paper flow diagram may be included here or in another section.

   Note: Actual forms are optional but their content is required.
d. Processing done by the System - Detailed information on how input data is validated, edited, and manipulated by the system and the user. All calculations performed by the system are shown in this section.
e. Outputs - What outputs will be presented to the user and on what time schedule. Actual printouts are not necessary but clear representations of the output contents must be included. Examples are a good technique--NO PRINTER SPACING CHARTS ARE PERMITTED!!
f. Control Totals - How the team and the user can be sure the system is working. Controls typically include sight verification by the user, internal program counters and control files containing various counters reflecting previous program executions. See the Leeson book, pages 228-230 for further details.
g. A Glossary of Terms - The definition of any terms NOT made clear in the body of the specification. These terms may be terms common to the user. (Use FOOTNOTES to refer to the glossary)
h. The Planned Implementation Schedule and Acceptance/Rejection Criteria - How the system is to be implemented

*"the system" includes automated and manual procedures.

(refer to Chapter 11 of the Leeson book); what criteria the user will use to determine if the system performs all aspects of the design agreed to in the functional specifications.

i. User Sign-Off (optional); Team Sign-Off (required)- The signatures of the user and all team members agreeing to the content of the specification. This is necessary before the functional specifications are handed in for grading on or before 3/15/83. Each member agrees to basic design--not necessarily the way the design is presented.

NOTE WELL: The Leeson book does not cover a document called a "functional specification." Chapters 3 to 6 do contain material to be included in the functional specification, as I have described it. Another excellent source of information on the functional specification is the book: Gildersleeve, Thomas R. Successful Data Processing System Analysis. New Jersey: Prentice-Hall, Inc., 1978. (Chapter 6) This book is on reserve for this course at the LCCC LRC in the instructor's name. Ask for it at the Circulation Desk.

4. One team member is responsible for submittal of a User Manual on or before 4/19/83. A user manual is a document describing, in detail, in "non-computerese" terms:

a. How forms are filled out.

b. How reports are interpreted.

c. All specifics any user of the system needs to know about what the system does. It must be specific enough that any user can read it and be able to use the system without any further instruction.

A suggested format is as follows:

a. General Introduction to the Manual - If there is a specific way the manual should be used, it must be described here. For example, a new entire manual in a specific order, a user with problems would be instructed to reference specific sections.

b. Table of Contents - A detailed list of all sections and their locations within the manual.

c. Section(s) on each input document or requirement and a listing of all codes. Complete instructions on the collection and recording of all data must be included. If fields are left-justified, right-justified, allowed to be blank, required to be filled in, optional, etc., it must be stated in this section.

d. Section(s) on each output report showing in detail: the report layout, what the source of each field is (calculation, input document, etc.), when each report is produced in the system and who receives copies of each report. Do NOT use printer layout charts in this deliverable.

e. A Glossary of Terms - Any terms that should be made clear to a new user.

f. Control Total Validation Procedures - If controls must be validated by the user, how that validation is to be done.

g. How to use the APPLE II microprocessor - Any information the user needs to know to operate the machine, enter data, and run programs. Be brief and to the point. Only discuss very basic instructions.

Refer to Part I of Chapter 12 of the Leeson book as a reference.

5. One team member is responsible for submittal of Detailed Systems Documentation on or before 5/3/83. This document deals with the "Computer" aspects of the system. It is the document given to a programmer from which all programs are written. It includes:

a. A concise overview of the entire system in narrative and system flowchart form.

b. Detailed program specifications - A specification for each program in the system including: narrative overview, input layouts, output layouts, detailed processing description (step-by-step procedure of what the program is to do). No program flow-charts are required.

c. For each file in the system, a cross reference list must be included to show, for each field, what programs modify, edit, or report it.

The same team member is responsible for writing one program to be demonstrated in the formal presentation. The program may be one actually used in the system or a special one demonstrating some aspect of the designed system. Note: it must pertain to the system.

A PERT chart for the project must be developed by this individual. A hand prepared version is required on or before 3/15/83. A computerized version is required using software available from the instructor on or before 5/3/83. A user manual for this software is on reserve in the Library - ask for "A.P.M. User Manual."

**Refer to Chapters 9, 12, and Appendix I in the Leeson book for information on this deliverable.

B. 4 Person Teams

The first three requirements are the same as for a 3 person team except that the third person is not required to write a program or develop a PERT chart. The fourth team member is responsible for writing and testing all programs in the system in BASIC according to the Detailed Systems Documentation. This person must be prepared to demonstrate all programs. The deadline for this is 5/3/83. This team member is also responsible for producing a PERT chart as described above.

NOTE: Samples of these deliverables are on reserve in the library in my name. Ask at the Circulation Desk for EDP 203 reserved materials. IMPORTANT - These documents are from previous semesters and may not satisfy the requirements of this handout. They are examples of what students have done in the past and must be used as such. Be sure to follow this handout for the requirements of your deliverable.

III. THE FORMAL PRESENTATION (maximum time 20 minutes)

All members of the team are required to participate in some aspect of the presentation. Each member's presentation should be approximately equal in length. The team may choose any approach to presenting the system. One used in the past that has proven very effective is to present the system and assume your audience is the prospective user group. Trying to "sell" your system usually makes for a more interesting presentation. The presentation is not the time to go into great detail about the system. You must strive to educate and convince your audience as to the merits of the system without boring them. Techniques for effective presentations will be discussed in class. I may invite guests to your presentations. Your presentation will be videotaped for review by the team members at a later date.

Samples of presentations from previous semesters are on reserve for this course in the instructor's name in the Listening/Viewing area of the library (2nd floor). They present a variety of techniques used in the past. You will find some techniques very effective and others ineffective.

IV. GRADING

All deliverables must meet the goals of the system.

Written Documents -

Criteria:

Meeting the deadlines - each class day (Mon.-Thur.) late is penalized by one letter grade.

Form - all deliverables must be typewritten, double-spaced with
- standard margins
- spelling, grammar, and punctuation mistakes will be penalized
- neatness is important - I expect college level work
- the length of the deliverable is not important-the content is
- no hard cover binders will be accepted - put a staple in the upper left hand corner of each deliverable or place in a soft cover 3-hole folder.

Content - the deliverables are to be written in a clear and understandable manner. I expect you to use what you have learned about written communication in this course in your writing.

Formal Presentation -

You will be graded on how well you present your material. You should make every attempt to make your presentation as interesting professional as possible. Presentation techniques are discussed during lectures on Chapter 4.

Each team will be evaluated by every other class member (anonomously) as they present. Each team will be given the results of these evaluations after their presentation. These evaluations do NOT count toward your grade in any way. They are intended as a feedback mechanism only.

Your Project Grade - (½ of your final grade for the course)

consists of

50% written document grade (includes weekly status reports)

There will be 13 status reports collected in total. Each is worth 1 point. Grading is based on neatness and the fact that they were turned in on time. Late or missing reports will be penalized 1 point. Other deliverables will be graded and returned 1 week from the date they are due except in cases of class cancellation.

50% formal presentation grade

NOTE WELL: Missing a class or leaving early on the night your team or another team presents will be penalized:

If yourteam is presenting - "0" grade for your part of the presentation.

If another team is presenting - 5 points deducted from your FINAL COURSE GRADE unless you have a valid, documented excuse.

Time will not allow for switching of presentation dates once they are assigned.

MANAGEMENT OF DATA: THE CATALYST OF INFORMATION SYSTEMS

Herman P. Hoplin

School of Management
Syracuse University
Syracuse, New York

ABSTRACT

This paper presents a way of looking at what makes an Information Systems curriculum a strong force that will be helpful to all persons working together to train for the new technology. For some years now, the introduction of the new information technology has been the subject of controversy, complexity, and change.

Data Management is at the heart of that whole issue and challenges our ability to use computer systems as tools of the manager or the user. For that reason, we need a good understanding of where we are today and whether we are ready in educational institutions to do the research and teaching needed to adequately educate the potential managers and users of this new technology.

For starters, the management of data is viewed as the catalyst needed to bring this all about. The issues facing managers of data are looked at, some interim solutions and methodologies considered, and finally, the objectives and strategies for Management of Data courses are suggested.

INTRODUCTION

The Information Systems profession is rapidly expanding both in scope and in depth. It is now truly interdisciplinary. No longer is it limited to computer science and traditional management. Communications, Management Science, and Behavioral Science (including Organizational Development) are vying for equal consideration in the evolving new technology. Even before these multiple problems of complexity and change entered the information systems scene, organizations such as DPMA's Education Foundation were concerned with the educational preparation for people entering the Information Systems profession. Due to these and other related curriculum development steps, it is now widely recognized that Information Systems are a vital component of the curriculums of collegiate schools of business at both colleges and universities and at both the undergraduate and graduate levels.

Why are Information Systems a vital part of the business curriculum? Surely not for using the computer as an end in itself. Successful business managers have raised questions such as, "What is the cost of one wrong decision?" It could mean a missed opportunity or at the other extreme, the downfall of a company. These managers are well aware of the value of information and, when properly informed, want the competitive advantage of having a computer-based information system to provide them with the tools and information to make the best possible decision. Information Systems development, the way it should be taught, can provide improved productivity by exploring both behavioral and systems techniques:

- This is not to say that managers have not always used available information to aid them in making decisions. It is only to recognize that with the advent of data processing that the capability now exists to process much more information on a timely basis than ever before.
- With the development of data processing, a dilemma now exists; paradoxically, the manager may be flooded with data but starved for useful information; or he may be suffering from an information glut where the system has not filtered out non-essential information for that particular manager.
- Further evidence of the need for Information Systems courses for business students was presented recently in a Harvard Business Review article by Brandt Allen[1] where he emphasized that many businesses are discovering that success of their Information Systems cannot be realized by technology alone. He points out that:

  > Senior managers have confidence neither in their ability nor in that of DP management to manage information resources. . . . Of all the important functions of a business, information systems is the one area where senior management most often lacks experience and understanding.[2]

It should be emphasized that this paper is not dedicated to one or more course experimental techniques or strategies in the effective teaching of a data management type of course. Rather it is the synthesis of years of research and experience. As such, it represents a "living" document intended to enhance the reader's understanding of an evolving need for continued development in the highly volatile and dynamic field of information systems. However, Figure 1 is a Basic Teaching Model for Management of Data Courses which is viewed in this paper as the catalyst that ties together an Information Systems curriculum.

VISUAL TABLE OF CONTENTS AND STRUCTURE OF PAPER

Visual Table of Contents (VTOC)

Direct-Access Device. This paper is a direct-access device. The reader may prefer, however, to initially go through the paper sequentially. Subsequent reference to a particular section of the paper may be made directly to the models or section apropos to the desires or wants of the person using it. Since the paper is user-oriented, an easy access path (PATH I) follows:

PATH I

1	INTRODUCTION
2	VTOC
3	ISSUES FACING MANAGERS OF DATA
4	SUCCESSFUL MANAGEMENT OF DATA METHODOLOGIES AND STRATEGIES
5	IMPACT OF MICROCOMPUTERS ON MANAGEMENT OF DATA
6	DATA ADMINISTRATION
7	OBSERVATIONS/PREDICTIONS FOR DATA MANAGEMENT
8	HIGHLIGHTS AND CONCLUSIONS FOR MANAGEMENT OF DATA COURSES
9	REFERENCES

Research Models, Diagrams, and Tables

These are reflected in the following listed diagrams and tables (PATH II) which are arranged in the order referred to in the paper:

PATH II

Figure 1	BASIC TEACHING MODEL
Figure 2	MIS MODEL
Figure 3	HIERARCHICAL CONCEPTUAL DATA BASE MODEL
Table 1	AREAS OF DA/DBA RESPONSIBILITY
Table 2	PYKE'S LIST OF QUALITY FACTORS

Content of Paper

Techniques vs. Substance. This vignette on the data management aspects of Information Systems is enriched with considerable substance obtained from enterprises in both the private and public sectors of the nation. The need to accelerate the development and delivery of data management information to both undergraduate and graduate students who will soon be in the workforce is of paramount importance. Nevertheless, academic approaches and techniques must be closely related to the real world situation and cannot be constrained effectively to an academic environment devoid of business and industry applications and experience. This paper with its teaching and MIS model on the one side and its concentrated coverage of the major issues facing managers of data on the other is intended to provide realistic views on how practical teaching techniques can be arrived at in an academic situation.

Data Administration. This major element of Information Systems is covered in considerable depth. The more successful ways of handling information through the use of the Data Base Concept is elaborated upon and the broadening role of the end user as the beneficiary of the systems is stressed. A model (Figure 3) is developed which shows the managerial implications and "value-added" benefits that can accrue from this approach.

Human Resource Aspects for Quality Information Systems. The person part of the so-called man/machine system is emphasized particularly from the point of view of the needs for academically-trained systems developers and data managers for facilitating organizational and systems growth from the point of view of the enterprise. This includes managers, end users, and technicians. Both internal and external environmental factors as well as quality of life become of vital importance for motivation and productivity.

Dynamics of the Field of Data Management. Rapid changes and developments in the field have justifiably forced a heavy long-range planning element into the life cycle of Data Base and other data management systems. This means that long-term commitments have to be made in enterprises which must also be reflected in Information Systems courses and curricula. Some trends and predictions are provided together with some guidelines on course development. In this connection, the Basic Teaching Model for Management of Data courses (Figure 1) is intended to broaden the horizons of both the new and experienced academicians by viewing the intended course as a problem with several major elements, each of which would be addressed by the student from a galaxy of seven interdisciplinary elements, These are required for success in an information systems course. The model shows the nature of the new technology as being multi-disciplinary, rapidly changing, and of growing importance to business and industry.

General Terms/Key Words

General Terms. *Design, Documentation, Human Factors, Information Systems, Management, Management Science, Microcomputers, Standards.*

Key Words. *Critical Success Factors, Data Base Administrator, DBMS, Data Management, EDP, Information Center, Model, On-Line Systems, Quality Improvement, Strategic Planning, User, Work Station.*

ISSUES FACING MANAGERS OF DATA

Information "Glut" Plus

Is it the information "glut" or is it a myriad of other issues that sometimes cause managers to wonder whether they are managing or being managed? Against a backdrop of a very short (30-year) period for developing a new computer-based information systems management profession, researchers such as Warren McFarlan have addressed several critical and recurring questions. These questions were raised by managers about their information systems activity and have been translated into manageable trends for the 1980's. They are listed below:

Six Manageable Trends Applicable To Information Services(IS)[3]

1. IS technology impacts firms in different ways strategically. The thrust of this impact strongly influences the appropriate selection of IS management tools and approaches.

2. The merging of office automation, telecommunications, and data processing technologies. Formerly disparate areas of technologies now requires coordination of at least at a policy level and later at a line control level.

3. Organizational learning about IS technology is a dominant fact of life. The type of management approaches appropriate for assimilating a technology change sharply as the organization gains familiarity with it.

4. Environmental forces are shifting the balance of IS services on make-or-buy decisions in the direction of buy, which profoundly impacts the kind and quality of IS support an organization can receive.

5. While the functions of the system life cycle remain, the best approaches to executing them, and the problems of implementation have changed significantly, with a wide diversity of approaches being appropriate for different systems.

6. Effective IS policy and responsibility involves a continuous reshifting and rebalancing of power between general management, IS management, and user management. Each group has a legitimate and important role to play in ensuring an appropriate level of IS support to the firm.

Self-renewal and Updating of Technical Matters

So much of the six themes listed above that pertain to management of data courses must be included in the syllabi for these courses if educators are to meet the issues facing data managers.

Additionally, careful selection of technically-oriented topics must be enforced to insure discrimination between need-to-know and nice-to-know hardware and software coverage for information systems. For example, as brought out by McFarlan[4] the understanding of the programming challenges of the rotational delay of the drum of an IBM 650 is of little value in meeting the challenge of today's information systems technology. Conversely, perhaps IS should take the lead in encouraging the greater use of structured-programming concepts to insure that IS logic is followed in solving a problem and so that others schooled in structured techniques can read programs developed by others. A recent editorial in InfoWorld addressed this problem referring to it as the dilemma that "Too few programmers are trained in the ways of 'structured programming.'"[5] Early work on the structured programming concept was done by Dr. Harlan Mills who has become known for the "Chief Programmer Team" concept.[6]

A Checklist for Managing Information

There are many so-called "Checklists" for managing information. In fact, each organization that is consciously aware of managing information has either a formal (written) or an informal one. In an article on data management relationships, two professors of Defense Systems Management College came up with the following:

A Checklist for Managing Information[7]

- Who generates information in our systems?
- Who switches the information?
- Who segregates the information?
- What about data loss?
- What is this information costing?
- Are the right people on the circuit?
- What about timeliness?
- How much can be routinized?
- How much can be delegated?
- Need-to-know versus nice-to-know?
- Bottlenecks in flow; human or system?
- Built-in feedback?
- What about corporate memory?
- Is there a recall system?
- What is the critical path of our information?
- Is this a push or pull system?
- What is the critical 20%?

Advanced Concepts To Think About

Without further elaboration below is a selected list of essential elements that are now becoming of great concern to Data Managers and as such deserve prominent places in data management courses of IS curriculums:

1. The greater use of On-Line Systems
2. Data Base Applications
3. Teleprocessing, Communications, Networking
4. Strategic Planning and Top Management Strategy
5. Work Station Idea
6. Information Center Concept
7. Human Factors
8. Power shift from a Central EDP office to the User Domain

SOME SUCCESSFUL MANAGEMENT OF DATA METHODOLOGIES AND STRATEGIES

Selected Frameworks for the Management of Data

Data Planning and Enterprise Model.[8] James Martin provides the objectives, advantages, and criteria for selecting data bases considering the types of data environment (files, application of data bases, subject data bases, and information retrieval systems). Martin uses top-down planning techniques for the planning of data systems which he analyzes a number of ways. (For example, routine versus non-routine processing and systems).

IBM's Business Systems Planning.[9] BSP is a program that provides a structured approach to assist a business in establishing an information systems plan. It is designed to appeal to top management and to satisfy the near and long-term information needs of the organization.

Critical Success Factors.[10] John Rockart introduced this method in an article in the Harvard Business Review entitled "Chief Executives Define Their Own Data Needs." This approach is based on the identification of "Critical Success Factors" in terms of information needs for individual managers. It is a valuable approach to information systems design since the MIT research team (of which Rockart was a member) found a high correlation between the CSFs and the attainment of the organization's goals. The method is highly effective in helping executives in defining the most important information needs to do their jobs in the company. This might be considered a management by exception method of systems design; nevertheless, it provides the key indicators needed for information systems development and numerous supporting areas crucial to the effective management of data.

Greater Use of Management Science in Teaching. The contribution of management science to information systems is becoming more and more important as the new technology develops and becomes more sophisticated. Industry heads of operations research departments such as Gerald Hoffman[11] advocate a deeper involvement of management scientists in information systems activity. Hoffman, in an article on this subject, contends that management science can magnify the productivity of information systems and that it can make major contributions to decisions which are totally outside the purview of such systems.[12] In fact, he defines a management information system (MIS) as:

> The function of an MIS is to capture data from events in the real world, process the data into information useful to a manager and transmit the information to him in a timely and useful way. The hallmarks of a good management information system are consistency, reliability, and clarity of presentation.[13]

Information Systems Strategies for Quality Improvement. M. H. Schwartz, General Manager of Designtech Corporation, an early practitioner in the management of information systems, and a pioneer in developing the concept of MIS has conducted several workshops on management strategies for quality improvement. Schwartz in his consulting assignments has concluded that quality has become a major strategy of today's large organizations. He is now advocating improving the quality of information systems as a fundamental part of IS strategy. He terms Quality a pacing critical success factor without which IS success is not attainable. According to Schwartz, adequate quality assurance has the following beneficial results:[14]

- Reducing the high costs of low quality
- Quality concepts and practices in systems development
- The imperative need for integrating quality control into system development processes
- The equally imperative need for independent quality assurance
- Motivating developers and line managers
- Quality as a key to ADP credibility and user relationships

IMPACT OF MICROCOMPUTERS ON MANAGEMENT OF DATA

A Revolution Within a Revolution

The Potential Role of Microcomputers in Both Large and Small Organizations. Just as the environment was getting accustomed to the invasion of minicomputers into small business, the potential role of minicomputers in large organizations began to make itself felt with great intensity. Auditing firms such as Arthur Young & Company are noting that this has led to high level interest by managers responsible for overall information systems policy.[15] In noting the emergence of microcomputer technological advance and integration in large organizations, Rhoda Canter of Arthur Young ran an analysis of these trends against Richard Nolan's well-known theory of data processing growth in organizations (Harvard Business Review (March-April, 1979). She found that whereas Nolan's six stages of EDP growth remain valid, but because the new microcomputer technology is introduced by the users, the roles of the data processing professionals and the users are somewhat reversed. For example, user "pioneers" first acquire the micros in Stage I; it is not until Stage VI that the application portfolios of the organization formally approve and integrate the application.[16]

January 1983 Federal Software Conference Keynote. Dr. Ruth Davis stressed the importance that should be attached to how software and hardware are becoming one in the desktop and the microcomputer. In commenting that one is going to see the design of the software simultaneously with the design of the chip that we may for the first time get the needed quality control on software which has plagued us for many years.[17]

What's in a Name? The terminology of distributed data processing has been confusing to say the least. This has made more difficult the job of determining the impact of this trend on data management. Gerald Hoffman of Standard Oil of Indiana clarifies this somewhat in defining the terms:[18]

- When small computers began to proliferate they were given the name "mini."
- The advent of LSI and VLSI brought another order of magnitude decrease in size and price and led to the name "microcomputer."
- As the focus changed from size to location, the former microcomputers are now called desktop computers and home computers.
- More recently, the label has been changed again to personal computers--a long overdue focus on the user - (micro in size and in cost; maxi in capability).

Hoffman alludes to the impact by stating, "But the name 'personal computer' and the opportunity to use it productively for individual and private purposes should not obscure the further opportunity to use these machines effectively and efficiently for the organization."[19]

Information Processing Strategies are Crucial for the Future of Colleges and Universities

In spite of a backlash effect when new ideas are proposed (existing organizations protecting themselves against change) of ten colleges and universities participating in an EDUCOM study,[20] all ten organizations include in their information processing strategies for the 1980's:

- Decentralization. All the organizations are moving to a more decentralized information procession environment. This trend means that an increasing amount of information processing activity will occur outside of the centralized facility.
- Personal Computers. All the organizations have, or are formulating, plans related to the growing potential of personal work stations for students, scholars, and administrators.
- Networking. All ten campuses are involved in both local and national networking activities.

One of the common assumptions accepted by key individuals on each of the campuses involved in the project was that:

> Since most of the work in colleges and universities deals with words and images, not numbers, the greatest future growth in computing will be in support of individuals who have not been represented in the traditional community of computer users.[21]

This will have a great effect on the management of data throughout the balance of the 1980's and beyond.

A Challenge to Business Executives

While academe struggles to revamp its strategies for campus computing to accommodate the information systems requirements of the users, business executives have a similar, if not personal, challenge brought about by the advent of microcomputers. That is the insatiable user-demand to cut out the "middle man" and gain direct access to the computer. Users want answers NOW. Users want to control their own destiny and generally have budgets to assemble their own resources. Adam Osborne, a pioneer researcher on microcomputer architecture and the founder of a company marketing personal computers, recently said in an interview, "Executives will discover they can't avoid using computers--they'll just have to do it."[22] This statement may be prophetic and is indicative of the rapidly growing microcomputer industry. In fact, many executives are already using computers and are changing their organizations to provide more direct interface with the computer by operating on-line functions. This trend does not, however, relieve the information systems people from coming up with easy-to-use systems to manage data and provide information for decision-making purposes.

DATA ADMINISTRATION

> *We are reaching the stage where the problems that we must solve are going to become insoluble without computers. I do not fear computers. I fear the lack of them.*[23]
>
> Isaac Asimov

The Management of Data: A Management Decision

Management Perspective. The problem of how to manage data, as illustrated in the Basic Teaching Model (Figure 1) involves both management (organizational and behavioral) and technical issues. As indicated in the Asimov quotation, the complexity and volume stage that we are in today makes it almost unthinkable to attempt a satisfactory solution to the management of data without the use of computer technology. But the management issues are the more difficult and therefore require careful planning followed by an implementation strategy. Most importantly, it is a management decision--not a technical decision that must be made to determine the future course of how an organization's data is going to be managed. Stating it differently, strategic planning for data management is an executive responsibility.

Management Information Needs. In deciding the strategy for managing its information, Management must determine what its needs are for data to produce the life-blood information for the organization. This involves careful planning, analysis, and design; consideration of the components necessary to make the information system effective and focused toward the organization's goals and objectives; the management structure needed to plan, develop, and ultimately manage the information system; but above all, it must develop a strong management commitment to the proposed strategy. These conditions are difficult (if not impossible) to accomplish unless the development effort is dedicated to meet the user needs and is developed against an environmental framework that the issues facing managers of data, some of which are discussed in a prior section of this paper. Figure 2 is a model of an integrated system that diagrams the various components that would be brought into juxtaposition with the environment envisioned in this paragraph.

Selection of Alternatives. Almost intuitively managers setting strategy and policy guidance for the management of data would opt for what might be called a "one-stop system." Managers, unlike students, do not have to be sold on the idea of the 'Systems Approach.' They do not like to manage fragments if the "whole" can be covered by a system.

1. As is implied in the lead quote by Asimov, computers store and retrieve large amounts of information. No longer can the clerk with a ledger and a quill pen of Charles Dickens' time do the job where vast amounts of data are involved. The division of labor set in and multiplied the clerk into many clerks. With the advent of mechanization, this manual effort became more efficient and soon voluminous transactions were batched which led to one of the clerks or machines performing one accounting function while other accounting functions were done by other clerks or machines. Electrical Accounting Machines (EAM) led to very large batches and the use of many trays of cards. This process was proliferated even more with the use of magnetic tape on computers. As James Martin pointed out, the use of data-processing techniques deprived management and its clerks of giving one transaction individual treatment and forced management to live with the compromise of data rigidly divided into separate files.[24] This compromise, according to Martin, has become the accepted method of operation with little thought given to its desirability.[25]

2. Management became more and more frustrated as it faced situations where its executives needed information that crossed departmental and sometimes company lines. It is no wonder that the traditional file system discussed in subparagraph 1 caused thinking managers to conclude that there must be a better way to manage data. Firms and organizations began thinking of a one-stop system that would allow its data to be processed as an entity or an integrated whole. This idea developed into the data base concept so well described by Richard Nolan in his now classic Harvard Business Review article, "Computer data bases: the future is now."[26]

3. With the exception of microcomputers and distributed systems, the data base concept, and the myriads of software products that ensued from it, was allegedly the top computer-related development of the 1970's. Firms and organizations have been converting to data base processing almost en masse, except at the small business level where diversity of data uses is not sufficient to justify this conversion.

4. Professional associations such as the Data Processing Management Association (DPMA) and the Association for Computing Machinery (ACM) have sponsored and are sponsoring many seminars and conferences on data base subjects. Computer-related periodicals regularly carry numerous articles on data base.

5. It is small wonder then that the demand for people schooled in data base is almost insatiable. To meet this need, Information Systems Curriculums, as sponsored by the Associations mentioned in subparagraph 4, are now including data base courses as the heart of the management of data educational effort. The balance of this section of this paper will be devoted to this subject.

The Role of Data Base Management and Education

Data Base Techniques. The subject of management of data can be adequately covered in data base oriented courses covering such areas as Data Base Concepts, Data Base Administration, Data Dictionary, Data Models, Data Base Management Systems (DBMS), Data Base Design, Data Storage and Access Methods, Data Base Performance, and perhaps some advanced topics such as Distributed Data Bases, Integrated Data Base Software Packages, Data Base Machines, and aids for data base performance evaluation and improvement. For purposes of this paper only, several of the areas listed will be covered. Except for some basic definitions and integration into the data base concept, detailed coverage of conventional file techniques is not required in courses designed to educate students in the management of data.

Why Data Base as an Educational Tool? As was brought out in the first paragraph of this section, it is a tactical management decision to determine how an organization's data is going to be managed. The data base issue is a strategic question requiring a strategic management decision. To answer the data base question requires consideration as to whether the company overall will be database oriented. Currently, this is a high growth area. Subparagraph 3 above indicates the trend to data base. Recruiters on campuses are looking for graduating students who are well-grounded in data base concepts. Therefore, this type of course(s) is a natural to build up the Information Systems curriculum.

What is Data Base? The references at the end of this paper plus an evergrowing number of data base books and articles provide a number of answers to this question. Since this is more of a procedural paper on the management of data than a substantive paper on definitions, this question will only be partially answered here. Perhaps what it allows an organization to do is more important than a simplified definitive response. It allows an organization's data to be processed as an integrated "whole." It forgets about individual files and stores data elements for all programs through association which eliminates redundancies and keeps the data independent from any one program to permit it to be used for all programs. Since it is interdepartmental, it crosses organizational lines and requires the sharing of the data by many users. In recent years, various security and integrity techniques have been developed to protect the data from unauthorized uses.

A Conceptual Data Base Concept

Hierarchical Associated Tasks (HAT) Model. Figure 3 illustrates the basic situation in the real world for the systematic management of data. The HAT model introduced is intended to be helpful in visualizing the change in concept or perception that takes place in an organization deciding to go data base. Such a management decision requires rigorous planning and the use of a model such as HAT may serve to clarify management's thinking on the issue.

1. Effective involvement of all layers of management is crucial for the success of the data base effort. Involvement is probably not a strong enough word; it is more of a management commitment. HAT is structured on the foundation of three layers of management (top, middle, and operations) which are related to the general tasks that each of those layers of managers performs.

2. The brim of the HAT depicts a data base which not only collects data elements needed for operating the business but transforms these elements into information for the use of the entire organization. The overriding thought here, in addition to the sharing of data horizontally and vertically throughout the organization, is the difference between

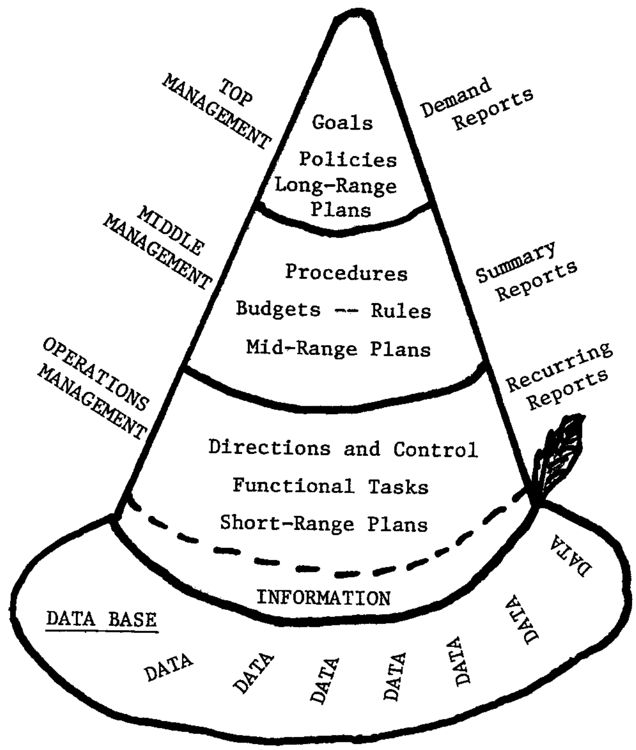


FIGURE 3: Hierarchical Associated Tasks (HAT)

the logical and the physical organization of data. The logical aspects are the concerns or the ways in which the users (managers and their functional areas) view the data. The physical aspects are of concern to the computer technicians who must be able to store, retrieve, and otherwise manipulate the data to be responsive to the users need for information. Simplicity is of primary concern to the user who may have little concern for the efficiency of the physical storage as long as he can get the information he needs on a timely and accurate basis. The software hides the complexity from the users although the technicians responsible for the physical organization face the constantly changing organizational and data migration situations that develop. The design goal of the data base should be to make managers more effective --not necessarily more efficient; the reverse may be the case for the data base specialists or technicians where complexity may actually be more advantageous in designing efficiency into the system.

3. The HAT model also shows the "value-added" benefits emanating from the data base into the various levels of management. Some hidden benefits turn out to be integrity and the ease of reconciliation of the information (i.e., reports) since using a data element only once is of great assistance in this regard, and management's ability to readily

access the data base by posing ad hoc questions needed to meet unprogrammed requirements. The first attribute mentioned is not infallible since the one data element introduced into the system could be wrong--but at least it is consistent which is better than having several varying elements and trying to reconcile them. The second attribute not only provides for accessing the data from many angles, but it contributes to providing the manager with prompt responses needed for decision making.

The Development of Data Base Technology

Data Base Models and Data Base Management Systems (DBMS). Starting from a zero base in the 1960's, the data base industry has propelled itself into a multi-billion dollar a year business. James Martin's comment made almost ten years ago, "The 1970's is the decade of the data base,"[27] may, indeed, have been conservative. He made this comment in noting that, "One of the most badly needed courses in universities, business schools, and other establishments which teach computing is a course in the realities of data-base technology."[28]

Beginning with an experimental system, ADAM (Advanced Management Data System) developed by the MITRE Corporation for the U.S. Air Force in the early 1960's and continuing to the present as represented by DBMS products from a host of vendors (IBM, Cullinane, Cincom, Software Ag, and Intel/MRI to name a few) we are now in an environment replete with selection possibilities.

Academically, we study these as being classified into several basic models that are not dissimilar from file access methods. The more common models are Hierarchical, Network, and Relational. There are, of course, many variations of these and many ways of evaluating them. The type of DBMS selected is based upon the requirements of the organization to optimize processes most essential to its success. Some trends are now taking place. These will be discussed in the following section of this paper.

Data Base Administration. The people part of data base systems, as in any man/machine system, is vital to the success of a data base application. These functions and responsibilities are generally vested in an organization headed by a "Data Base Administrator (DBA)." Some data base technical writers break this down into Data Administration which refers to non-technical managerial activities such as planning for the data base and the responsibilities for the data and its definition; and the DBA who is responsible for the decisions and the operations of the data base. Since the major data base responsibilities revolve around the combined position, it is sometimes better to teach it as a single-headed organization. In any event, a function of major importance is the relationship between the DBA and the user. The DBA must be able to translate user requirements or requests into action through the technical side of his/her organization. For a capsule of what the job entails Table 1 provides a listing:

TABLE 1: Areas of DA/DBA Responsibility[29]

- Data Definition
- Data Base Design and Implementation
- Data Base Access
- Data Base Standards Control
- Documentation
- Operations
- Monitoring
- Data Base Management System Enhancements
- Education
- Vendor Enhancements

Data Dictionary. In the early years of data base applications, data dictionaries were non-existent. It was discovered that since the DBA has the responsibility of defining the data, the procedures, controls including standardization, and who is responsible (the owner) of the data items that a tool was needed to help perform these vital functions. Hence, the development of the data dictionary, sometimes referred to as the data dictionary/directory. It is also a data base design tool that is important to the user. Not only does it serve to enforce the agreements reached on data items and structures, but it keeps the user informed of authorized changes and gives him a ready reference for many questions that would otherwise have to be referred to the DBA. The Data Dictionary is now a major component of virtually all commercial DBMS.

Classroom Exposure to Data Base Systems

1. Some data base texts advocate hands-on exposure to a data base system or to data base applications. Unless conditions at the university or college are favorable for a hands-on experience, this is sometimes the hardest part of setting up for a data base course.

   a. If the university or college itself uses a data base, the use of the data base in the classroom is greatly facilitated. A data base in being that can be used in the classroom (or at least as a demonstration) solves many pedagogical problems.

   b. An even better situation exists where the school within the university (i.e., School of Business Administration) has a dedicated DBMS in a computer center where students can access instructional material (information) in the data base. This arrangement usually cannot be built over night, but rather over several semesters. Within this time-frame, it can be well established and become readily available for course use.

2. In schools where a data base is not in use, the problem is more severe. On at least one occasion, the instructional data base recommended in the text would not work on the university system. In another case, the vendor's demonstration was so inadequate that even the most basic questions on data base could not be answered. Other cases of quickly trying to get a data base application for educational purposes degenerated into retrieval systems that have very awkward software for operation under the data base concept.

3. Schools that do not have a data base in operation have several options open to the "use" of a data base in the classroom:

a. First, establish a "user friendly" relationship with the computer center (academic or administrative) so that the technical knowledge of that group can be utilized in arriving at a solution.

b. Then, lay plans for some sort of information (or laboratory) center where graduate students or doctoral condidates can work on the problem. If students in the data base class are assigned to that activity as a project, the work should be progressive (i.e., each succeeding class picks up where the prior one left off).

c. Use guest speakers who are either specialists or consultants in data base or have capabilities to put on a demonstration (either in the classroom, in a local company, or at a vendor's office).

d. Assign the students to panels to study certain aspects or various steps in data base, the conversion to data base, or on the evaluation and selection of data base management systems or other tools. It is not unusual, particularly in a graduate class, to uncover students with experience on various DBMS who can assist in arranging speakers or on-site demonstrations.

e. Continue the search for an instructional data base that is available, will work, and is up-to-date in representing the state-of-the-art as it exists today.

4. A word of caution on all of these possible solutions. Be sure that the hands-on or practical exposure is not counter-productive by having the students wait in long queues to gain access to a terminal or the "university" system. A motivational situation can quickly be reversed by students spending hours trying to get on a terminal, getting a message that the system is down, or by trying to find a syntactical or keyboard error. The experience must be positive and should be focused at educating information systems students--not architects or technicians.

OBSERVATIONS/PREDICTIONS FOR DATA MANAGEMENT

Managers and Data

An Exercise in Self-realization. Warren Schmidt stated recently in An AMA Survey Report, "Managerial Values and Expectations:"

> Managers' optimism about the future varies according to how much control they feel they have.[30]

As information systems specialists, we can conclude that the data or information that the manager uses is a mirror image of what he considers the most important; and that he must feel in control of his company's or department's destiny. To have this control, the mechanism for establishing policy and priority must rest with management and not the data processing or some other department. A further conclusion can be drawn that the control of data management no longer rests solely with the department processing the data, but is now shared with the using organization(s)

Use of "novel" Organizations. The situation summarized above will lead to the increased use of flexible, project-related, or interactive-type organizations. Some of these are:

(1) Project Management matrix-type organizations to develop information or data base systems.

(2) Information Centers staffed by MIS and EDP departments with specialists trained to assist users in developing their own systems.

(3) "Open Shop" EDP Departments with only a top-level skeleton staff of specialists, analysts, and consultants. The decentralized units of data processing will be relocated in the user departments.

(4) Work Stations distributed physically into operational environments or executive offices.

Quality User Service. To facilitate the process of the user assuming responsibility and ownership of systems, the quality of the work environment must be built through increased credibility and trust throughout the entire organization. The end result of this is to ensure that the system (or the data base as the case may be) performs exceedingly well in serving the user. To do this requires attention to the factors listed in Table 2.[31]

TABLE 2: Pyke's List of Factors Associated with Quality User Service

1)	Functional fidelity--A user must be sure that functions performed by the system are accurate.
2)	Timely execution--Response times in online systems and throughput in batch must meet user needs.
3)	Ease of use--Access to, and use of the system must be reasonably easy.

4) Reliability--The mean time between failure and mean time to repair must be within acceptable limits. This includes backup and recovery of the data base and its software as well as hardware components.

5) Availability--Data must be accessible.

6) Upgradability--The system must provide the ability to increase the amount and type of service.

7) Privacy and security--Legal and economic aspects must be considered in the protection of the data resource.

Management and Human Factors. Data Processing now is only one part of a much broader spectrum that includes communications, office automation, distributed data bases, interactive software, and a host of other exciting developments. Management will have its own systems (i.e., MIS) and they will be computer-based, but certainly not EDP department-oriented or managed. The traditional EDP shop has to change drastically if it is going to maintain any kind of desirable status or image in a business organization.

Data Management

Data-Base Technology will become the Backbone of Most Data Processing.[32] This caption is a statement by James Martin of a prediction that is already taking place. He points out that, "It is unlikely to succeed fully unless the end users it serves are intimately involved in certain aspects of its design."[33] This prediction of what is to come speaks for itself except perhaps to emphasize again that unless conventional data processing operations change to permit normal user growth that the users will continue to make significant inroads into what has been the domain of data processing.

Data and Information Sharing. This is basic to the data base concept and definite trends are running in this direction. It should be noted that this often requires changes in management philosophy and in the organization because, in the final analysis, the data base(s) can only work successfully to support a management organization that is ready and willing to share and control its information in this manner. Welke and Konsynski, in discussing this view, point out that there has been a time lag of nearly ten years in arriving at a data management user system as being something other than existing computerized files. They make a convincing statement that, "If one cannot 'see' a DBMS in the user system, it is difficult to design one."[34] As this concept becomes better known (partially through data base courses) user demand for converting to data base will become even greater than it has been so far.

Distribution of Information. This relates directly to the earlier section of this paper on the impact of microcomputers on the management of data. With the possible exception of the effect of data base technology, the distribution trend will result in the greatest change in data management for the future. In fact, its influence is also forcing the design and development of distributed data bases with many accompanying implementation and control problems. This development is increasing the users' sphere of control and simultaneously is relieving the EDP department of many mundane and iterative-type functions that will wind up at the user 'remote' sites.

It is only fair to state in discussing what amounts to decentralized data processing that problems do exist because of the difficulty to control and set policy to givern such operations. However, once this is done the users will take up the cudgel to plan for and develop management information systems that are useful to their organizations. The streamlined corporate EDP organization will have to set a realistic course with respect to user involvement and render technical assistance--not resistance to change.

Development of Data Base Management Systems (DBMS).

1. Referring back to the earlier discussion of data base models and DBMS, it is apparent now that Relational Systems will dominate the late 1980's and 1990's. Despite some problems now in the handling of massive transactions, relational concepts are "clearly superior to existing DBMS technology and vendors like IBM's long-range objectives are tied to the success of the relational concept."[35]

2. Data Base Machines to eventually supplement or replace software DBMS packages are in various stages of research and development. Some vendors now have transition systems of marketing what amounts to a dedicated CPU to run the accompanying software DBMS.

3. DBMS vendors are quickly becoming applications software vendors where the DBMS is the major member of an entire family of software designed to expand data management capabilities. Data Dictionary, report writers, and security systems are a few of the many user tools now offered. This will expand and change the scope of what is currently referred to as "The DBMS Marketplace."

Data Base Administration Career Field. Data Base Administrators and Data Base Specialists have taken their place among the top positions in the Data Processing field. They are in great demand in large and medium-sized installations. Percent of pay increase for starting salaries rose from 7-11% from 1981 to 1982.[36] With the high demand for microcomputers and distributed data processing it is likely that smaller installations and user departments will swell the demand for these data base personnel to an even greater demand.

HIGHLIGHTS AND CONCLUSIONS FOR MANAGEMENT OF DATA COURSES

TEACHING DSS TO MBA STUDENTS -- EXPERIENCES WITH AN EXPERIMENTAL COURSE

Lois Graff

Associate Professor of Management Science
George Washington University, Washington, D.C.

An experimental course in Decision Support Systems and the results of that course are described. The course is offered through The George Washington University School of Government and Business Administration for MBA students. The general goals are to acquaint the functional manager with the opportunities presented by Decision Support Systems and the difficulties which may attend the development of such a system, and to suggest ways in which the functional manager can effectively participate in the process of design, implementation and use of such a system.

Increasingly, persons who are not Information Systems specialists are becoming directly involved in the use of computers in their business activities. This increased involvement takes the form of actual hands on computer use, selection of computer hardware and software for personal use in making business decisions, and in participation in the design and development of such end-user systems.

Many factors have contributed to this trend. Among them are advances in software technology leading to packages which make it relatively simple to construct or modify "user friendly" Decision Support Systems, advances in hardware technology leading to the proliferation of microcomputers within the purchasing range of individuals and small organization units, a continuing backlog of application requests in many data processing departments, and increasing computer "literacy" among today's managers.

The growth of end-user computing has many implications for the Data Processing department. Viewed in a positive light, this increased participation can reduce the burden on the Data Processing department and increase user satisfaction. This requires, however, that the functional manager, who is not an information system specialist, be educated to participate effectively.

Toward this end, the Management Science Department of the School of Government and Business Administration of George Washington University, offers an introductory course in Decision Support Systems for students in the MBA program. The course was offered for the first time in Fall, 1982, and is directed toward students whose goals do not include becoming specialists in Information Systems. Its only computer prerequisite is an introductory course in data processing. This paper discusses the design of the course and summarizes some of the experiences gained. The presentation will discuss these experiences further and will highlight some of the difficulties associated with the course as well as its accomplishments.

OBJECTIVES

The explicit objectives of the course are to develop an understanding of:

- what a DSS is,
- the process of designing and developing a DSS,
- and the role of the functional manager in the DSS process.

It is difficult to define precisely what a DSS is, or to exactly demarcate MIS from DSS. Perhaps a very broad definition might be a system which can integrate a variety of data manipulation and modeling tasks on an ad hoc basis and which is easy for a non Information Systems specialist to use. To expand this definition and provide the student with a better understanding, the various areas which are frequently used to define DSS are explored in the course. Thus, discussion includes such topics as:

- how the application areas associated with DSS differ from those generally associated with MIS/EDP,
- what the characteristics of a "good" DSS are,
- the issues currently confronting designers and users of DSS,
- and possible future directions for DSS development.

To gain an appreciation for the development process of a DSS, the course looks at:

the differences between the prototyping development process for DSS and the system development life cycle of traditional EDP/MIS,
how user input can be obtained, evaluated, and used in the functional design,
how to develop a DSS design structure which provides easy adaptability,
what is acceptable documentation for the DSS,
and, the development and delivery of user training.

Because the primary audience for the course is the non Information Systems specialist, it is particularly important to try to develop a notion of the role of the functional manager in the DSS process. Certain topics seem to be particularly relevant to this purpose. These are:

the development of functional requirements for the system,
the evaluation of the user/system interface in selection of a package which can generate Decision Support Systems,
evaluation of system documentation,
and evaluation of the quality of and requirement for user training.

COURSE STRUCTURE AND REQUIREMENTS

The class meets a total of 14 times including the final exam. Each regular class meeting is divided into a lecture/discussion and a laboratory session. The lecture/discussion period centers around readings in the text book (Decision Support Systems, an Organizational Perspective by Keen and Morton, published by Addison Wesley) and assigned readings (see bibliography).

During the course the student is directly exposed to the Interactive Financial Planning System (IFPS) by EXECUCOM and the Visisystem (Visicalc, Visitrend, and Visiplot) through class demonstrations and assigned problems. These are used to demonstrate such concepts as procedural vs. non-procedural languages and multidimensionality. Other commercially available systems are discussed and compared. Several case studies are reviewed and used to illustrate the concepts discussed in class.

The most critical part of the course involves the development of a personal DSS on a microcomputer. The class is split into teams for the project, each team to develop its own DSS. The laboratory part of the class time is for team work on the DSS. This provides both time and space for the large (ten to twelve students) groups to get together and for the instructor to review their progress reports with them and to discuss and help them with any difficulties. The groups are required to present their completed systems to a group of outside observers at the end of the semester.

In addition to the group work on the DSS project, each student is required to write a brief issue paper. The topics may explore such issues as how to gain acceptance and increase use of a DSS to the evolution of a DSS application into a MIS application. For this paper the student is required to research current thinking in the area, and then to critically analyze the issue in terms of his/her own insights.

The final class meeting is reserved for class evaluation and final exam. At this meeting, the instructor and students share observations on the learning experiences. The final exam, which is the only formal testing in the course, is an essay type designed to force the student to integrate the reading material with the learning which occured in the group project.

THE CLASS OF 1982

The Students

The course is offered for MBA students. For the most part, the students in the Fall, 1982 offering of the course fell into this category. A few, however, were doctoral students and a couple were non-degree students. The students tended to have fields of concentration either in information systems management or in finance, though there was a fair representation from many other fields.

Student ages ranged from mid 20's to mid 40's. Some were full-time students, but the majority were part time students who were fully employed elsewhere. Some had had no exposure to computers beyond an introductory course which included some BASIC programming. Others had worked for several years in computer related areas. In short, the class was quite heterogenous with respect to experience and knowledge.

The DSS Project

Some critical elements of the course structure were borrowed freely from Klein * with adaptations made because the course offered in the MBA program has less stringent data processing prerequisites. In particular, the specific topic used for the class development of a personal DSS in Fall, 1982 is from Klein. This was a DSS to support the purchase of a new car by an individual. The students were expected to investigate the decision processes which occur in this situation and to determine which of the decision points could best be supported by an automated system. They were then expected to design the DSS and to implement it on a microcomputer (a 48K Apple II microcomputer with two disk drives was used).

The class project helps the student to see both sides of the system development process (user, professional developer) by playing both roles. The class was divided into two rival teams to simulate a competitive market environment. Each team was further subdivided into three groups, project management, pre-design (functional design), and design (structure of software system design).

* Theodore Klein, Boston University, chair of panel discussion: "DSS Curriculums, Courses & Teaching Methods," DSS-82, San Francisco, July, 1982.

The project management group was responsible for overall progress of the development, for developing and monitoring time schedules, for planning the use of time in the weekly class laboratory sessions, and for locating and allocating resources. It also had primary responsibility for the formal end of term presentation. The project management group was required to submit a biweekly report on progress.

The pre-design group was the primary contact with the intended user of the system. It was the task of this group to investigate the decision process, identify the points at which decisions are made, and to determine where automated support might be helpful. These decision points were then put in priority order by the pre-design group. For the most useful decision support points, the group described functionally what the system might do for the user. This included specifying the kind of data which might be useful and the kinds of verbs or models which might be used. The pre-design group was responsible for determining the feasibility of obtaining the data desired.

The pre-design group then negotiated with the design group to determine which and how many of the relevant support systems could feasibly be developed within the time frame given. The design group was responsible for designing the actual software system. The software development and system implementation was shared by the entire team under the direction of the design group. The design group was charged with developing documentation and any user training required.

The last session of the course was a market presentation of the products to a potential group of users, persons who had experience in the purchase of automobiles, but who varied in their knowledge of computers. These users then rated the two systems and noted their strengths and weaknesses. These comments were summarized and discussed with the students.

It was hoped that the addition of an element of competition would provide incentive to develop a system which really did address the problem from the perspective of the typical user. In fact, it seemed to be extremely difficult for the groups to continue giving sufficient weight to user needs when faced with the difficulties of implementation. Compromises were made between the initially agreed upon functional description and the design of the final product which ultimately weakened the products in both cases.

Additional Learning Experiences

The primary objectives of the course deal with introducing the student to Decision Support Systems and exploring the role of the functional manager in their design and development. The nature of the group project, however, provided the student with other learning experiences which are at least as valuable. First of all, the need to organize in order to accomplish a task in a brief time forced the student to try to anticipate the tasks which had to be accomplished, to use some basic techniques of project management, and to identify, locate and allocate resources required.

The team effort itself yielded some excellent examples in group and organizational dynamics. Not only did the individuals within each subgroup have to learn to work together, but the groups within a team had to negotiate with each other in order to win support for their ideas. The quality of the final product could be directly traced to the degree of group cooperation.

The end of term marketing presentations not only provided incentive to develop a good product, but also allowed the students the opportunity to prepare such a presentation. They had to be able to describe and demonstrate their products briefly without resorting to jargon and to respond to questions from the assembled "clients." This presentation was a very positive experience for most of the students, both in terms of obtaining the satisfaction of being able to show their substantial efforts outside of the class confines, and in obtaining comments on the advisability of some of the compromises which had been made.

Finally, many of the concepts which were discussed and practiced in the course can be extended directly to apply to EDP/MIS applications. Thus, while the non-technical user is ostensibly learning about Decision Support Systems, s/he is also coming away with a better appreciation for the difficulties of system design and implementation in general. We hope this will produce managers who can participate more effectively in decision making for information systems in the future.

FUTURE DIRECTIONS

This initial offering of the course in Decision Support Systems has convinced the author that this is an area rich in a variety of learning experiences. It is quite clear, however, that the course would not have been successful if there had not been persons in the class who were technically qualified. Given the scant prerequisites for the course, this was a fortunate serendipity. It will be a challenge to future offerings to find a way to incorporate both the functionally and the technically competent.

The selection of a topic for development of the DSS was somewhat unsatisfactory. Due to the lack of functional prerequisites, a project was selected which was generally part of the knowledge base of most students. While this was valuable in trying to get the project going, the topic of automobile selection was criticized as not being appropriate for computer support because sufficient non-computerized support was available.

Given the large section of the class which was majoring in finance, there was some desire to see more financial applications. This raises the much larger question of course content and objectives. This course was specifically designed to teach about DSS to non-technical persons. There seems to be a great deal of desire, however, for courses in the functional areas which incorporate the use of the computer in a much more substantial fashion. This is a trend which we are seeing more and more, however, until we have faculty who are both knowledgeable and interested, and universities with sufficient computer resources, hardware, software and support, large scale incorporation of the computer into non-computer courses will continue to be a challenge.

BIBLIOGRAPHY

Text Book

Keen & Morton, Decision Support Systems, An Organizational Perspective, Addison-Wesley, 1978.

Readings

Alter, Steven L., "Gotaas-Larsen Shipping Corporation: A Corporate Planning System," from Decision Support Systems, Current Practice and Continuing Challenges, Addison-Wesley, 1980.

Dearden, J., "Myth of Real-Time Management Information," Harvard Business Review, May-June, 1966.

Dearden, J., "MIS Is a Mirage," HBR, January-February, 1972.

Dempster, Michael A.H., "Issues for the Future in DSS," Summary, Proceedings of the International Institute for Applied Systems Analysis, Pergamon Press, 1980.

Dickson, Gary, chairman, "Issues for the Future in DSS," discussion group 1, Proceedings of the International Institute for Applied Systems Analysis, Pergamon Press, 1980.

Gerrity, Thomas P., Jr., "Design of Man-Machine Decision Systems: An Application to Portfolio Management," Sloan Management Review, Winter, 1971.

Hackathorn, Richard, chairman, "Issues for the Future in DSS," discussion group 4, Proceedings of the International Institute for Applied Systems Analysis, Pergamon Press, 1980.

Keen, Peter, W., "Decision Support Systems: A Research Perspective," Proceedings of the International Institute for Applied Systems Analysis, Pergamon Press, 1980.

Keen, Peter, "A Policy Statement for Managing Microcomputers," 1982, unpublished.

Little, John D.C., "BRANDAID: A Marketing-Mix Model, Part 2: Implementation, Calibration, and Case Study," Operations Research, July-August, 1975.

McKenney & Keen, "How Managers' Minds Work," HBR, May-June, 1974.

Methlie, Leif, chairman, "Issues for the Future In DSS," discussion group 2, Proceedings of the International Institute for Applied Systems Analysis, Pergamon Press, 1980.

Simon, Herbert A., New Science of Management Decision, Harper & Row, 1960.

Sprague, Ralph H., "A Framework for Research on Decision Support Systems," Proceedings of the International Institute for Applied Systems Analysis, Pergamon Press, 1980.

Wagner, G.R., chairman, "Issues for the Future in DSS," discussion group 3, Proceedings of the International Institute for Applied Systems Analysis, Pergamon Press, 1980.

VIGNETTES OF INFORMATION SYSTEMS TEACHING TECHNIQUES: HUMAN RESOURCE MANAGEMENT

Kathryn M. Bartol

University of Maryland

David C. Martin

American University

Various trends suggest the need for greater concern with the effective management of human resources within the information systems function. Information systems instruction must place increased emphasis on three major areas aimed at enhancing human resources: human resource planning, human resource development, and interpersonal skill development. Teaching techniques related to each of these three areas are suggested.

INTRODUCTION

Effectively managing human resouces has become a critical challenge to the information systems field today. Traditionally, the field has placed great emphasis on technology; but, more recently, has focused increasing attention on interfacing appropriately with the rest of the organization.[9] However, the area which has continued to receive inadequate attention is the effective internal management of the information systems function and particularly the management of human resources--the people. The effective management of human resources has become a critical issue for several major reasons.

First, people costs have been skyrocketing relative to hardware costs. Wage and fringe benefit costs have become seemingly insatiable drains on information systems department budgets.

Second, recent studies show that many of the problems associated with information systems design and implementation ultimately can be traced back to personnel and internal management issues, rather than to technological shortcomings.[17] The linkage is becoming more obvious between the quality of the human resources in the information systems function and the quality of the systems developed.

Third, the critical shortage of qualified information systems personnel has delayed the optimum use of available technology. Many projects which could be developed wait in cues while potential users grow inpatient. A danger is that with the growing availability of small computers and specialized software, users will be encouraged to implement systems on their own rather than interface with seemingly unresponsive information systems departments. Some distribution of information systems capability is both desirable and inevitable; but, carried to its extreme, the organization can easily lose control of its critical data.

Fourth, Moore notes the existing "disparity between the ability to create sophisticated and reliable hardware technology and the ability to develop and implement software systems useful to management."[13] As available software improves, the emphasis in information systems is gradually shifting from a concentration on inventing new software to a greater focus on creative applications of existing software. This trend requires a different approach to human resource development than has been commonly used in the past because different skills are needed.

Fifth, the role of information systems specialists is unique. Like engineers, they need technical and administrative expertise to some degree. But they must also be able to take the language of upper level management, translate it for hardware and software use, manipulate the data as required, and subsequently give upper level managers the information that they need to make critical decisions. As top management becomes more acutely aware of the vast potential of the computer for aiding their decision making, the job of the informations systems specialist is destined to become even more complex.

As a result of these trends, information systems instruction needs to place more emphasis on three major areas related to the internal management of information systems departments: (1) human resource planning, (2) human resource development, and (3) interpersonal skill development. The purpose of this paper is to discuss instructional needs in these three areas and to suggest appropriate teaching techniques.

HUMAN RESOURCE PLANNING

Planning Issues

Human resource planning has been uncommon among information systems departments. Instead, most information systems managers have relied on the current labor market to provide skilled personnel when needed. This approach frequently has proven inadequate--leading to such drastic measures as offering "bounties" to employees who help recruit other information systems experts. In particularly good times economically, this keen competition generates a great deal of personnel movement--much of it disfunctional.

Willoughby estimates that annual turnover in the data processing field ranged between 15 and 20 percent during the 1960's, declined to about 5 percent in the early 1970's and began to rise again by the end of the decade.[16] A recent Datamation study placed the current annual turnover rate among information systems personnel at about 28 percent annually. At this rate the equivalent of one's entire staff turns over in a little better than three years.[12] Furthermore, it is quite likely that the best people are lured away. Nevertheless, the ability to buy quick technical expertise, although frequently at very high cost, has enabled many information systems managers to hide a lack of personnel planning and the existence of inferior personnel practices which would not be tolerated in other parts of the organization.

Two related factors suggest that relying on a market personnel staffing strategy will eventually lead to more severe difficulties than in the past. Nolan has argued that as information systems departments grow, they need to move through six stages toward maturity if they are to perform effectively. Movement through the maturity curve involves a fundamental change in the role of the information systems manager from management of the computer in the early stages to a later stage concentration on the management of data resources. The later stages include a much heavier involvement in facilitating improvements in the strategic position and upper level decision making of the organization.[14]

We have argued elsewhere that movement along the maturity curve will require personnel with different characteristics than are commonly found in information systems departments.[1] As has been the case, personnel will need to have state-of-the-art technical knowledge. However, they will also need to acquire extensive knowledge of the particular "business" of their organization in order to help determine where the use of computer power can best enhance the efficiency and effectiveness of that organization. The real challenge is that this combination of technical and organizational expertise cannot easily be bought; rather it must be built.

In addition, projections regarding the future of programming predict a fundamental change in the role of programming personnel. Wasserman and Gutz suggest that the availability of flexible software will eventually shift much of the effort of programmers over to applications work where they will need not only technical knowledge but also added expertise about the operations of the organization.[15]

These developments suggest a need to think in a more strategic way about the types of personnel that will be needed in the long run. This notion is tied into a more fundamental requirement--the need to develop a strategic plan for the information systems department which interfaces with the overall strategic plan of the organization. Stated in another way, the information systems department needs to plan so that it will be in a position to carry out the information systems aspects of the strategic plan of the organization. Indeed, in the age where timely information and relevant, flexible information systems increasingly can mean the competitive edge, the ability of the information systems function to strategically plan will be crucial to its success. The strategic plan must ask:

Where do we want to be?

Where are we now?

What do we need to do to reach our goals?

A major aspect of strategic planning must be aimed at having qualified people available to do the necessary jobs at the appropriate time.[7] Hence, human resource planning asks:

What types of people and skills will be necessary to carry out our strategic plans?

What types of people and skills are available now and are likely to be available given current hiring, attrition and retention rates?

What can be done to make up any shortfalls in needed people skills?

This latter issue will be discussed later under the heading of human resource development.

Training Approaches

The major point here is that we need to integrate strategic thinking and basic human resource planning into information systems classes. There are a variety of ways to accomplish such integration.

One way is through the use of written cases depicting particular data processing departments and their changes over time. These could be used by students to analyze the adequacy of strategic directions and planning, as well as the human resource planning done by the department in the case.

Another alternative is to use hypothetical organizations that are similar to real organizations or departments. In this approach, students can be asked to consider likely strategies and personnel needs in order for these organizations to maintain and enhance their competitive positions.

Another method is to develop written case situations where students must decide among alternative personnel to staff planned projects.

If the situations are realistic, there are likely to be considerable tradeoffs and shortfalls in the skill levels available for optimum staffing of the projects. In analyzing these types of situations, the problems associated with inadequate staff skills become very obvious and students can begin to think in terms of how best to avoid such situations. These situations can be parts of cases where students are asked to develop information systems. That is, human resource planning issues can easily be made part of cases presented primarily for the purposes of technical training for the students.

Still another approach is to use live situations, where students analyze an ongoing information systems department or an organization in the industry. Here they can actually develop strategic plans and do the overall human resource planning. Frequently this approach can be accomplished through an outreach vehicle such as student practica.

For reasons mentioned earlier, we believe that the exclusive use of the market staffing strategies of the past will lead to disaster in the future. Of course, it will never be possible to fully anticipate the needs of the future or to predict future staffing conditions with total accuracy. Nevertheless, planning greatly increases the likelihood that staff with the necessary capabilities will be available when necessary. Because it will be necessary to build staff capabilities for the future, there is a need to train students to engage in human resource development.

HUMAN RESOURCE DEVELOPMENT

Human resource development focuses on methods for bringing about a match between the available human resources and those needed to implement strategic plans. Several of the main approaches are discussed here.

Development Methods

There are a number of different approaches which can be used to develop human resources. Those which appear to be particularly relevant to the information systems environment are job analysis, selection, career planning, job design, training, and performance appraisal.

Job Analysis. Knowing the specific requirements of each information systems function is necessary if individuals are going to be developed to perform more substantive and complex tasks. A human resource development program should be based on some form of a building block concept where the primary skills of the lower level job are reflected in some form in the next higher job in the organization. These primary skills become the base knowledge an individual carries to a higher level position. They assist the individual in mentally traversing the area from the known to the unknown in a new job. Specific information systems functions are easily recognized in a job description which has been prepared from thorough job analysis.

Selection. Thinking about human resource development should begin before new staff members are ever hired. Under the current market staffing approach, personnel frequently are hired to, at best, fit immediate needs. Little thought is given to how well the individual will meet the long-run needs of the organization.

A staff member who is hired to meet a short run crisis may have only limited capabilities for development in the directions necessary to be in synchronization with strategic plans. While it may be possible to encourage this person to leave at a later date when the individual is no longer valuable, such staffing strategies cause unnecessary personnel turmoil and are extremely costly in terms of recruiting and training costs, to say nothing of the lost opportunity costs. Lost opportunity costs are concerned with the fact that it may have been possible to hire an individual with good long-term potential who, through experience and development, would now be an invaluable asset from a human resource point of view.

Career Planning. Once a new person has joined the organization, it is important to engage in career planning. The main purpose of career planning is to develop employees who will be effective in terms of the long-run plans of the organization and the information systems function. The notion of career planning also implies the involvement of the employee in the planning process, so that the planned development meets individual as well as organizational needs.[8]

Career planning also helps to reduce the possibility of obsolescence. The problem of obsolescence is particularly acute in the information systems field because of the rapidly changing technology. Unfortunately, obsolescence tends to occur very gradually. Without career planning which is updated periodically, the obsolescence may well go unnoticed until the individual has reached the point where the obsolescence is very difficult to overcome. Such occurances often are as much a failure of the human resource development process of the organization as they are a failure of the individual involved.[7]

There is another benefit to career planning. Research strongly indicates that information systems professionals, like professionals in other areas, are motivated by achievement, interesting work, and growth opportunities.[4] Enhancement of professional opportunities through career planning strengthens commitment to the organization. In the absence of such career opportunities, positions in other organizations--often at twenty percent higher salaries--begin to look very attractive.

Job Design. Another means of building growth opportunities and human resources is through job design. There is evidence both within and outside of the data processing field which indicates that growth opportunities offered by the work itself have important effects on job satisfaction and work capabilities.[1]

Some writers have argued that we are in danger of making data processing related jobs so narrow as to stifle individual development.[10] This is particularly a danger with programming; but other areas are susceptible. For example, there is a tendency to keep individuals working on the same system over long lengths of time because they know that system. In a sense, they become part of the "indispensible subordinates syndrome," where individuals are rewarded for their good work by being stuck on a particular assignment far beyond its developmental potential. It is important for managers and those so affected to understand the long-run costs of this strategy in terms of human resource development.

Training. Training is a cornerstone of the long-run development of human resources. The need for training in a field characterized by rapid technological growth should be obvious. However, many organizations think that they are saving money by keeping training costs to a minimum. They do not count all the direct and indirect costs of minimal training. By not keeping their people trained, they are forced to continually fill in deficits in technical expertise by hiring people at high salaries on the open market.

At the same time, their high performers also are doing some analysis of their own. Seeing that their longer term prospects are severly damaged if they stay in a job which does not keep them at the forefront technically and with little prospect of keeping up with technological change without training, they begin looking for better opportunities elsewhere. Soon they become turnover statistics. The organization that they have left must now go through the expense of recuitment, selection, and orientation. Even if the organization finds another high quality employee, it will likely be at least six months to a year before the individual becomes fully productive because of the need to learn about the new organization.[12]

Breaking this cycle is not easy. Both information systems professionals and their employing organizations have become accustomed to the market cycle. Career planning begins to break the cycle by helping individuals analyze their strengths and weaknesses in relationship to present and future skill needs. The next step is developing a plan to alter any gap between present and desired skills. In fields characterized by rapid change, training is a necessary part of the plan.

Because information systems organizations are moving along the maturity curve and because the nature of the skill requirements is changing, it is likely that much of the future training will need to encompass not only technological issues, but also focus heavily on business knowledge. Furthermore, managerial skills are increasingly needed.

Performance Appraisal. Performance appraisal helps employees and their managers assess the degree of gap between expected and actual development and performance. Ample research supports the notion that feedback is important to the process of individual development and that it is related to performance.[3] For this reason, performance appraisal also is a powerful developmental tool in the information systems environment.

Without some form of performance appraisal it is difficult for information systems professionals to know where they stand in terms of the performance and developmental expectations of the organization. Because performance news is not always positive some managers avoid performance appraisals and falsely assume that employees know "how they are doing."

Training Approaches

Because the future of the information systems field is so dependent on human resource development, it is important to help students understand how the developmental tools available can be used in an information systems environment. They should have some familiarity with these tools from management, organizational behavior, and possibly personnel/management of human resources courses. The challenge now is to help students recognize how these developmental tools are applied. There are a number of instructional approaches that can be used.

In the previous section on human resource planning, we discussed the possibility of having students analyze cases or live organizations, develop strategic plans for the information systems department, and engage in human resource planning. Students could then be asked to determine the methods which should be used to move from where the organization is in terms of human resources to where it should be based on their human resource assessments. Their developmental plan could address particularly issues of selection, career planning, job design, training, and performance appraisal.

To address job design issues specifically, students could be asked to organize a hypothetical information systems department, including the design of jobs. They could then analyze those jobs in terms of the technological as well as career implications.

A similar approach would be to have students analyze jobs in a current organization in terms of the technological and career implications. This would help students gain knowledge about how work is organized in industry and aid them in understanding the managerial implications of job design in the information systems field. The students could be asked to go a step further and analyze the current training available in terms of its adequacy in developing employees to meet future skill demands.

Students could interview several information systems professionals on the topic of careers in information systems in order to gain insight into the options available and how working professionals see career development issues.

Students could use actual performance appraisal methods from local information systems departments to assess work on their own group technical project done in conjunction with class. They then could analyze the performance appraisal methods in terms of their usefulness in human resource development and in increasing employee effectiveness on the project.

INTERPERSONAL SKILL DEVELOPMENT

As mentioned previously, human resource development is aimed at assessing overall personnel needs in relation to strategic plans. Career planning helps assess the specific development needs of individuals. While many developmental needs obviously are particular to individuals, there are some skill needs that are fairly universal. For example, current research indicates that interpersonal skills are extremely important in information systems work.

One survey of important managerial skills found that information systems managers in 35 organizations rated generalist skills such as people and organizational skills ahead of technical skills such as systems and computers.[2] Similarly, another survey of data processing managers which sought the qualities required for effective managerial performance found that ability to relate to others was ranked first.[6] It is clear from these and other related studies that strong interpersonal skills are important for information systems managers.

While it is relatively easy to recognize the need for good interpersonal skills at the managerial level, there also is ample evidence from research on successful systems implementation and user relations which indicates that good interpersonal skills are important among information systems professionals in general.[1] A large part of the work of the information systems professional involves interacting with both users and fellow professionals.

Interpersonal Skills

Interpersonal skills take time to acquire, yet they are needed immediately by new entrants to the field of information systems. The need becomes greater as these individuals move on to increased responsibilities. Because of the rather generic need for basic interpersonal skills, it is advantageous to afford students practice within information systems programs. This can be done within the context of their technical training, as well as in specialized courses aimed at managerial issues related to information systems. Here we discuss two particularly important interpersonal skills, communications and group interactions.

Communications Skills. Wasserman and Gutz, in their study of the future of programming predict that the long-run need will not be for programming skills as we now know them; but, rather "for the ability to communicate with users and to understand the needs of these individuals and their organizations."[15] In the area of information systems, there already is ample evidence that the success of most information systems is dependent in part on the ability of information systems specialists to accurately understand the requirements of the user and to convey to users the enormous potential of the computer in meeting those requirements.[11] Two basic communications skills that could be easily taught in conjunction with technical materials are the orientation toward listening and the ability to summarize what one thinks one has heard.

The orientation toward listening involves training students to ask questions and let the users do much of the talking during the fact-finding phase of designing a user-oriented information system. Looked at in another way, the idea is to dampen the tendency to "tell the user how it is," particularly before understanding the situation.

Training in the ability to summarize is aimed at having the information specialist repeat back to the user his/her understanding of the situation. This helps the specialist check that he/she does indeed understand what the user said or meant to say. It often also helps the user clarify what he/she meant to communicate. This type of communication checking, especially at the early stages of a project, can prevent horrendous difficulties later in the project.

Group Dynamics. Training future information systems specialists to work in groups also is important. Much of their working life will be spent working with user groups and with specialist groups within the information systems department.

In addition, many of the new "innovations" in the management of systems development involve groups. Examples are chief programmer teams and structured walk-throughs. Ideally, students should have some experience working within these types of group configurations, as well as project management, before leaving an information systems program.

Training Approaches

To build communcations and group skills, one approach would be to provide user situations where some members of the class could take the roles of users and others the roles of information systems specialists (or each student could play a user in one group and an information systems specialist in another to get both perspectives and to even out the work load). The idea would be to have the information systems specialists get the necessary information from the users and prepare a proposal for an information system which they must present to the user group for reaction.

Another approach would be to set up some cooperative effort with industry where students could present proposals for information systems. Perhaps a situation could be described to the students, with opportunity for questions. Then different student groups or project teams could develop proposals for information systems would would be judged by managers of the organization. The students could then critique their own groups in relation to factors which facilitated performance and factors which hindered performance.

Students could be given a fairly major class project and could then be asked to propose an organization for their group which they would operationalize while doing the class project. This approach would provide experience in organizing themselves and working within the group.

These approaches also could be tied to the job design and performance appraisal developmental issues mentioned in the previous section.

CONCLUSIONS

The curriculum recommendations which have been made recently by the DPMA Education Foundation Committee on Curriculum Development give explicit recognition to the need for instruction in information systems management.[5] The recommendations give particular emphasis to communications skills--skills which we would argue are basic needs of information systems specialists as well as managers.

The spirit of what we have in mind here is captured to some degree in the comment at the end of the write-up for one course, "Structured Systems Analysis and Design." It is recommended that students do a semester long case study type project. The comment then notes that "students should engage in the process of analysis and design instead of merely concentrating on the products." While recognizing the need to cover a great deal of technical material within an information systems program, we also suggest that there are numerous opportunities to have students address human resource issues in conjunction with their technical work.

In summary, we have argued that students must be taught to think strategically in terms of where the information systems department or informations systems function is headed. There should be an information systems strategy which is tied to the strategy of the rest of the organization. Where the organization comprises strictly information systems functions the organizational and information systems strategic plan will coincide.

Once there is a strategic plan, then human resource planning must take place. The era of the current market approach to staffing is coming to an end for organizations that want to be in the serious competition when it comes to systems development and applications.

Once the human resource plan is in place, the next stage is to consider methods for implementing the plan. Major human resource development methods that we have discussed are job analysis, selection, career planning, job design, training, and performance appraisal. Students need to become familiar with the application of these techniques in an information systems environment. They will in many cases have had a general introduction to these topics in courses which comprise the American Association of Collegiate Schools of Business common body of knowledge requirements.

As it reaches the individual level, human resource development will involve the development of specific skills. We have focused on communication skills and group dynamic skills as being generic skills within human resource development. These skills are particularly applicable to the field of information systems because so much of the ultimate success of information systems rests on the ability of information systems specialists to both adequately communicate and effectively work in groups.

The possibilities to incorporate human resource management issues in information systems courses are endless. We have presented a number ideas to facilitate the creative process of instructional design in order to address human resource management within the information systems discipline.

REFERENCES

1. Bartol, K. M. and Martin, D. C. Managing information systems personnel: A review of the literature and managerial implications. MIS Quarterly, in press.

2. Benbasat, I., Dexter, A. S., and Mantha, R. W. Impact of organization maturity on information system skill needs. MIS Quarterly, 1980, 4, 21-34.

3. Carroll, S. and Schneier, C. E. Performance appraisal and review systems: The identification, measurement, and development of performance in organizations. Glenview, IL: Scott, Foresman, 1982.

4. Couger, J. D. and Zawacki, R. A. Motivating and managing computer personnel. New York: Wiley, 1980.

5. DPMA Education Foundation Committee on Curriculum Development. DPMA model curriculum for undergraduate computer information systems education. Park Ridge, IL: Data Processing Management Association Education Foundation, 1981.

6. Elliott, C. Qualities of a data processing manager. Data Management, January, 1975, 13, 35-37.

7. Glueck, W. F. Personnel. Plano, Texas: Business Publications, Inc., 1982.

8. Hall, D. T. Careers in organizations. Pacific Palisades, CA: Goodyear, 1976.

9. Keen, P. G. W. Information systems and organizational change. _Communications of the ACM_, 1981, _24_, 24-33.

10. Kraft, P. _Programmers & managers: The routinization of computer programming in the United States_. New York: Springer-Verlag, 1977.

11. Lucas, H. C. _Implementation: The key to successful information systems_. New York: Columbia University Press, 1981.

12. McLaughlin, R. A. That old bugaboo, turnover. _Datamation_, October, 1979, _25_, 97-101.

13. Moore, J. H. A framework for MIS software development projects. _MIS Quarterly_, 1979, _3_, 29-38.

14. Nolan, R. L. Managing the crisis in data processing. _Harvard Business Review_, March-April, 1979, _57_, 115-126.

15. Wasserman, A. I. and Gutz, S. The future of programming. _Communications of the ACM_, 1982, _25_, 196-206.

16. Willoughby, T. C. Computing personnel turnover: A review of the literature. _Computer Personnel_, 1977, _7_, 11-13.

17. Zmud, R. W. Management of large software development efforts. _MIS Quarterly_, 1980, _4_, 44-55.

LEARNING BY DESIGN: CONSTRUCTING EXPERIENTIAL LEARNING PROGRAMS

Daniela Weinberg and Gerald M. Weinberg

Weinberg and Weinberg
Rural Route Two, Lincoln, Nebraska 68505.

Abstract
How can trainers and educators assure that their students will really learn and retain ideas and information? In our own training efforts, we have found that people learn best through active interaction with their environment. Through direct experience with new materials, people construct new knowledge and make it their own. In this workshop, we present a theoretical framework for experiential learning. We outline a teaching methodology for applying this framework in a wide variety of learning settings. We also offer practical guidelines for designing and implementing experiential learning programs that really work.

How can we assure that our students really learn? If we want to improve the learning process, we'll have to decide what we mean by "learn." Recall a time when you learned something new -- a new skill, a new technique, a new word, a new hobby, a new way of interacting with people -- and consider the following three questions:

First, why did you bother? You knew that learning would change you, and that change meant stress. Why would anyone volunteer for additional stress?

Second. how did you feel during the learning process? Was it difficult or easy? Was it a delight or a bore?

Third, what was the outcome? Did your behavior change? Did it stay changed or did it fade? Would you consider it a successful learning experience?

Our most recent learning experience has been how to use a word processor. Let's consider Dani's answers to the three questions.

Why bother?

Jerry and his secretary, Judy, made a quick transition from the typewriter to the new machine. Dani observed the process and felt curious about this new addition to the family, as well as increasingly excluded by the new language around the office. Not to be conquered by a glorified typewriter, she asked for instruction.

How did you feel?

Dani's first reactions to the new technology were mostly negative. The keyboard was not laid out exactly like a typewriter's, nor did it have the same "feel." There was no carriage return, the familar typing sound was absent, and the small video screen had a nasty habit of displaying cryptic error messages such as:

62, FILE NOT FOUND, 00, 00

And with all her intelligence, she could not seem to memorize the simplest formatting commands but had to look up everything in the manual, over and over. No question about it, this was a stressful situation that made her feel lost, dumb, and angry!

What was the outcome?

With time and practice, Dani's irritation gradually gave way to excitement, which she noticed when she found herself actually looking forward to the next session at the machine. She knew she had mastered it when she realized that she no longer referred to the manual, or even to her little sheaf of penciled notes.

Piaget's Model of the Learning Process

How can we explain this process of learning? And, once explained, can we generalize the model to apply to other kinds of learning?

Jean Piaget, the Swiss philosopher and psychologist, proposed that learning was a process of constructing--not receiving-- new knowledge. The learner could not simply sit back passively, and obtain learning through the manipulations

Data Management--The Heart of the Information Systems Curriculum

This paper has developed the concept that the management of data is the central theme of an information systems curriculum--an emerging discipline to meet business needs by preparing qualified information specialists and systems professionals.

In order to meet the above conceptual objective, the major management-related content issues, elements, impacts, and observations were reviewed and associated to the classroom situation where such an association could be directly made to provide an example or suggest a teaching technique. Other issues in the paper with implied classroom applications together with other appropriate topics would be covered by course objectives and syllabi.

Many of the principal issues facing managers of data and some thought-provoking advanced concepts were discussed and followed by some suggested methodologies and strategies that have been successful in the business environment and in teaching.

Because of the overwhelming impact of microcomputers and distributed systems on the management of data, this subject was covered. The idea of personal or microcomputers at the senior management level and the easily-accessed information systems that their use makes possible makes this a "must" for inclusion in a data management or related course.

Data Administration

The heart of this paper is the section on Data Administration. The idea of how this is to be done is a management decision--not a technical decision. This is of cardinal importance to the success of any information system.

The emergence of the data base concept as a preferred alternative to the traditional file management system cannot be over-highlighted. The paper discussed the role of data base management and its use in information systems education. A conceptual model is presented to indicate the change in concept that transpires when an organization makes a decision to convert to data base management. This is followed by discussing several of the basic data base models and the development of generalized DBMS that are often studied in the classroom. Recognition is given to the people part of the data base system with a succinct coverage of the Data Base Administrator and his or her responsibility.

Classroom exposure to Data Base Systems and some of the problems faced in many universities and colleges is presented with a word of caution about some pitfalls that happen unless the application is well-designed and impeccably implemented.

Observations and Predictions

Manager (user) impacts discussed in the previous section are great resulting in powershifts from EDP departments to using departments and the use of "novel" organizations. But the results can be quality user service.

The more effective form of data management, data base management depends on a change in management philosophy to permit the sharing of data and information across departmental lines. Of the various models available, the Relational model is expected to dominate in the late 1980's.

Management of Data Courses

In terms of the DPMA Model Curriculum for Undergraduate Computer Information Systems Education[37] the following courses would fall generally in the Management of Data area:

- CIS-6 Database Program Development
- CIS-10 Decision Support Systems
- CIS-11 Advanced Database Concepts
- CIS-12 Distributed Data Processing
- CIS-14 Information Systems Planning

Guidelines to Use in Developing Specific Courses. Although not a comprehensive list, the following are the major guidelines to consider in developing or setting up an Information Systems course:

1. How does the course fit into the overall curriculum?
2. To what extent does the course overlap with or duplicate other courses?
3. What are the objectives of the course?
4. How are the objectives related to the competencies students will need in their subsequent academic careers; in their entering the business environment?
5. How are the various content elements related to the course objectives?
6. How is the course organized in terms of lectures, labs, panels, discussion sections, guest speakers, demonstrations, on-site visits, examinations, and other types of scheduled class sessions.

Hopefully, these guidelines and the other material in this paper will be helpful to the reader in the teaching of Information Systems and more specifically in the teaching of the Management of Data which binds together this emerging academic discipline. The challenge to excel and succeed in the developing and teaching of data management courses will greatly enhance *building the Human Resources for Quality Computer Information Systems.*

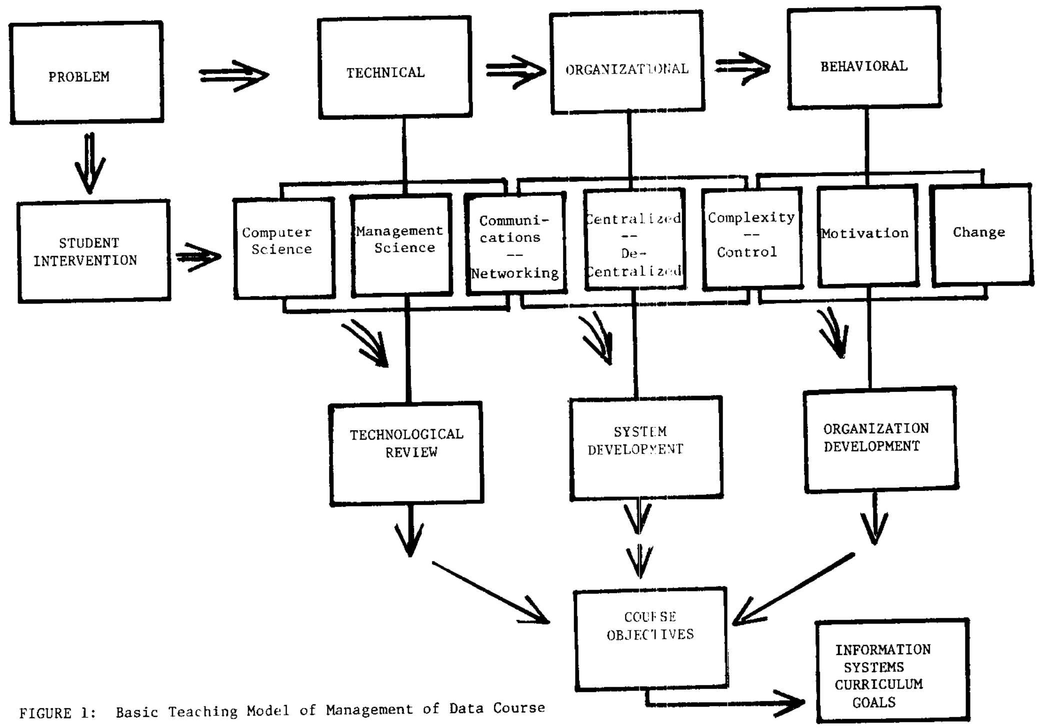


FIGURE 1: Basic Teaching Model of Management of Data Course

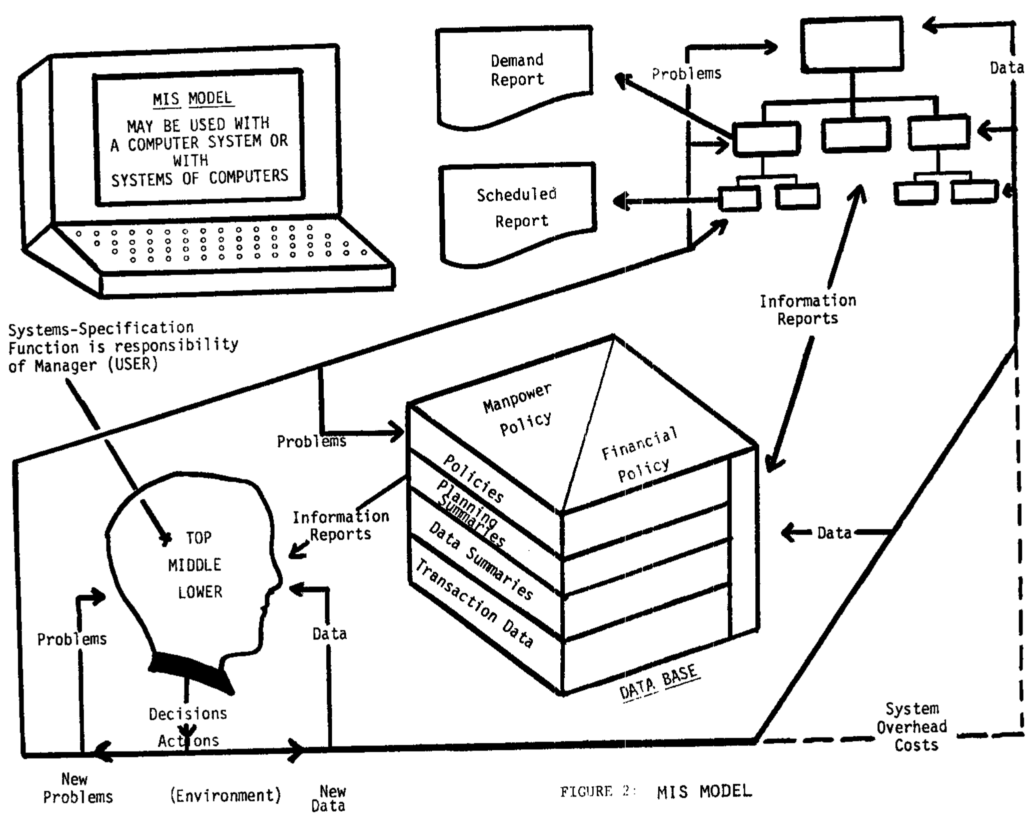


FIGURE 2: MIS MODEL

REFERENCES

1. Brandt Allen, An unmanaged computer system can stop you dead," Harvard Business Review (November-December 1982), 77.

2. Ibid., 79.

3. F. Warren McFarlan and James L. McKenney, Corporate Information Systems Management. Homewood, Ill.: Richard D. Irwin, Inc., 1983, 13, 14.

4. Ibid., 1.

5. Editorial, InfoWorld, October 18, 1982, "Pascal enrolls in frosh class," 34.

6. Dr. Harlan Mills, "Structured Programming," as quoted in Edward Yourdon, Techniques of Program Structure and Design. Englewood Cliffs, NJ: Prentice-Hall Inc., 1975, 83, 138.

7. "A Checklist for Managing Information," Program Manager, The Defense Systems Management College Newsletter (March-April 1981), 11.

8. James Martin, Strategic Data-Planning Methodologies. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1982.

9. Business Systems Planning--Information Systems Planning Guide, IBM Corporation, 1975, Preface.

10. John F. Rockart, "Chief executives define their own data needs," Harvard Business Review (March-April 1979), 81-93.

11. Gerald M. Hoffman, "The contribution of management science to management information systems," Interface (November 1978), 34-39.

12. Ibid.

13. Ibid.

14. Brochure for A Federal Information Management Forum held at The George Washington University School of Government and Business Administration on "Managing the Systems Function," Washington, D.C., February 14-16, 1983; M.H. Schwartz, "Management Strategies for Quality Improvement."

15. Rhoda W. Canter, "The Impact of Microcomputers on Information Processing Management," Government Computer News (December 1982), 1.

16. Ibid., 4, 5.

17. Dr. Ruth Davis, "Personality Profile," Government Computer News (December 1982), 3.

18. Gerald Hoffman, "Let's Not Get TOO Personal --OR--What's in a Name? Members Forum. Chicago, Ill.: Society for Information Management, No. 5, 1982, 2.

19. Ibid.

20. Donald T. Sanders, Editor, Perspectives in Computing, Vol. 2, No. 3 (October 1982); John W. McCredie, EDUCOM, Princeton, New Jersey, "Strategies for Campus Computing," 10.

21. Ibid., 6.

22. Jack B. Rochester, Editor, "Talking with Adam Osborne," An Interview, ComputerWorld Extra! (November 17, 1982), 20.

23. S. Atre, Data Base Structured Techniques for Design, Performance, and Management. New York: John Wiley & Sons, 1980, vii.

24. James Martin, "Corporate Data Bases," Principles of Data Base Management. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1976, 3.

25. Ibid

26. Richard Nolan, "Computer data bases: the future is now," Harvard Business Review(Sept-Oct 1973.

27. James Martin, Principles of Data-Base Management. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1976, xiii.

28. Ibid.

29. Auerbach Publishers, Inc. Practical Data Base Management. Reston, VA: Reston Publishing Company, Inc., 1981, 96.

30. "Highlights and Conclusions," Managerial Values and Expectations - The Silent Power in Personal and Organizational Life. Warren H. Schmidt and Barry Z. Posner, authors. An AMA Survey Report, American Management Association, New York, publishers, 1982, 9.

31. Thomas N. Pyke, Jr., "Assuring User Service Quality in a Distributed Network," Tutorial on Distributed Processing, IEEE, COMPCON 77 (Sept.'77) 5-32.

32. James Martin, Principles of Data-Base Management. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1976, vi.

33. Ibid.

34. R. J. Welke and B. R. Konsynski, "Technology, Methodology & Information Systems: A Tripartite View," Data Base (Fall 1982), A Quarterly Publication of SIGBOP, 44.

35. Ronald G. Ross, "Editor's Say-so . . . Predictions for Data Management: The Next Ten Years," Data Base Newsletter (November-December 1982), 2.

36. Robert Half, Prevailing Financial and Data Processing - Starting Salaries 1982. A study conducted by Robert Half on starting salaries since 1950, a pamphlet published by ROBERT HALF, Accounting, financial and edp personnel specialists.

37. DPMA Model Curriculum for Undergraduate Computer Information Systems Education. Prepared by the DPMA Education Foundation Committee on Curriculum Development, Data Processing Management Association Education Foundation, 505 Busse Highway, Park Ridge, IL 60068.

of another person--a "teacher." The learner was neither an "empty vessel" to be filled with knowledge nor a "blank slate" on which knowledge could be inscribed, but a system that learned through its own interaction with its environment.

What we like best about Piaget's model is the way it makes the learner an active participant in the learning process. The process begins when the learning is provoked--the learner examines his or her existing cognitive models because of some pressure from the outside environment. It might be provoked by necessity, as when a businesswoman struggles to learn Japanese to retain a major overseas client. It might be simple curiosity or, as in Dani's case, a feeling of being left out of the group. Whatever the case, learning doesn't just happen. Learning is the result of decision and action by the learner. (And not learning can also be a conscious decision and action.)

Once the decision is made, the learner plunges into an unfamiliar sea filled with unknown flora and fauna. At first, nothing fits or even makes sense in the learner's vision of the world. The world seems filled with cryptic messages. Piaget called this stage disequilibration. The learner is literally off balance in the new environment.

What to do? From the disequilibrated state, we may try to restore equilibrium without learning by changing the environment. When Dani said "this is nothing more than a glorified typewriter," she was bending the environment so she could retain her own model of the world. Piaget called this process assimilation, which was one pole on a continuum of choices we can make to restore equilibrium.

At the other pole, we can throw out our old model and embrace the new one totally. Piaget called this choice accommodation. Jerry is closer to the accommodation pole because he can no longer type competently on the typewriter without ruining 50 sheets of paper.

Generally, neither pole by itself provides a satisfactory outcome. Instead, we move between these poles in a dynamic equilibrium that Piaget called self-regulation. We partially surrender our earlier models of the world, and at the same time we twist the world into their static shape. In the process, we build for ourselves a new model of reality that somehow reconciles the old and the new. Although Dani no longer calls the word processor a typewriter, she still needs a printout and a red pencil before she can "really" edit her work.

When we complete the self-regulation process, we have constructed new knowledge. And then the process begins anew--moving from a state of comfort, into a provocation, through a state of discomfort, and finally into a new region of comfort.

The Learning Cycle

As trainers and educators, we can use the Piaget model to design teaching strategies that allow this process to develop. We must first put our students into a provocative environment. We must encourage them to experiment--to play with the materials in that environment. We must challenge some of their preexisting notions of the world--to provoke disequilibrium, but without utterly frightening them away. We must support and guide them through the self-regulation process. And finally, we must provide opportunities for them to consolidate and feel comfortable about their newly-constructed knowledge.

The teaching design we use to accomplish all this is called the learning cycle. The original learning cycle was developed by Robert Karplus of the University of California at Berkeley to assist students in the development of logical thought. Robert Fuller worked with Karplus at Berkeley and brought the learning cycle idea back to the University of Nebraska, where it was modified for college instruction by the ADAPT faculty. In our own work, we have adapted the learning cycle for adult learners. In all its variations, the learning cycle has three phases, called exploration, invention, and application.

The Exploration Phase

In the exploration phase, students--usually working in small groups--are permitted to explore a new environment on their own, with minimal intervention on the part of the instructor. The exploration phase is the learner's personal encounter with the new material.

Working with these materials, the students attempt to complete an apparently simple task. They soon find, however, that there is some fly in the ointment. Resolving the difficulty requires even more interaction with the materials, as well as active discussion with peers. Eventually, the students complete the task and possess a set of data about the new environment.

Here's an example of an exploration taken from one of our workshops for systems analysts. We want the students to learn a new way of designing questions, one that will be effective at extracting information from users in the fuzzy early stages of the system development process. We give them a work order for a new system, with instructions to prepare a list of questions for the user. The user is played by one of the instructors, who carefully answers their questions without volunteering anything not covered by their questions. The students are aware of some difficulties as they interview the user, but other difficulties become apparent only after they are given a complete list of all the information they might have obtained if their questions had been better framed.

The instructor's principal role in the exploration phase is designing a task that will force the students to call their existing models into question. Sometimes these tasks work too well, so the instructor may have to encourage a student who feels unable to proceed, but this is rare. Sometimes the instructor plays a role, as in our fuzzy-question simulation. Mostly, however, the instructor simply observes the explorations, gathering data to be used in the next phase.

Invention

The invention phase brings the entire group together, with the instructor playing a leading role. The group tries to make sense of the data generated during the exploration phase. They are now working in an analytical mode, trying to generalize from their data by inventing new concepts or tools. The instructor may provide the standard technical terms for these "inventions," or present a model in current use, or even deliver a mini-lecture to illuminate or integrate the students' inventions.

For instance, in our fuzzy question simulation, the instructor might guide the discussion by listing students' examples of information that their questions failed to elicit. Alongside each piece of information, the students write questions that might have succeeded. The instructor then asks the students to identify systematic differences between the successful and unsuccessful questions. Out of ten principles that the instructor had in mind, the students might develop seven on their own, get two more with tiny hints from the instructor, and need a bit of a push to catch the last. The instructor might then finish the invention by relating all ten principles to a model based information theory.

Regardless of the instructor's contribution, the knowledge that emerges has been constructed by the students themselves, rather than provided by the instructor. The instructor may have provided conventional names, or accelerated the convergence of invention, but has in no sense filled the empty student vessels with knowledge. That is done by the student's own active involvement.

Application

The application phase completes the learning cycle by creating the opportunity for the students to interact with the world once again--this time using their newly acquired models. As in the exploration phase, the instructor merely provides a structure and observes the students working in their small groups. In our fuzzy-question learning cycle, the application involves a work order for another system, more or less repeating the structure of the exploration phase.

In many cases, the application phase for one learning cycle becomes transformed into the exploration phase of the next. The fuzzy-question learning cycle is actually a series of case studies. In each case, the students progress a bit further by applying their learnings from the previous case--then run into trouble on the next type of difficulty. Such a series of linked learning cycles makes learning more efficient, but, more important, integrates different conceptual models.

Experience with the Learning Cycle

We have used the learning cycle approach in a great variety of educational settings, ranging from undergraduate college courses to intensive residential workshops for technical leaders in industry. Whenever principles are to be learned, rather than simple procedures, the learning cycle is appropriate and effective.

Virtually any subject can be learned using the learning cycle approach. Our experience, and the experience of our colleagues, has included anthropology, computer science, management, English, systems analysis, philosophy, logic, economics, communication skills, and leadership training.

We have used the learning cycle effectively in groups ranging from 5 to 150. Learning cycles may be designed for a variety of available resources, including what the students carry around in their heads. A learning cycle may be built into a one-hour class meeting or planned to run for a full 16-week semester or 6-day workshop. It may be integrated with readings, films, field trips, laboratory experiments, guest experts, and even lectures.

Learning versus Teaching

Piaget's model highlights the difference between learning and teaching.

In other educational approaches, the "teacher" acts on the "student," hoping to transfer knowledge from one to the other. As one wag described the lecture method, "It's a way of getting material from the teacher's notes into the student's notes--without passing through the brain of either one." Not all lectures for all students may share this characteristic, but it's certainly a frequently observed phenomenon--one that's simply not possible with the learning cycle. In the learning cycle approach, the material must pass through the student's brain because the student acts on and interacts with the material to be learned--transforming the material rather than mechanically copying it.

In the learning cycle, the student has responsibility for her or his own learning, which is virtually a requirement for success with adult learners. And through this responsibility, all students, whatever their age, develop a greater sense of ownership of the products of the learning process. Not only is understanding of the material much deeper, but retention is far better than in other approaches. Experiments show that most of what is taught is rapidly forgotten, but what is learned stays learned--until it's unlearned in some new disequilibrating environment.

References

SCIS, Teacher's Handbook, 1974
Lawrence Hall of Science
University of California
Berkeley, CA 94720

This is the original publication containing the idea of the learning cycle as developed by Karplus.

Piagetian Programs in Higher Education, 6th printing, 1982
ADAPT Program
110 Ferguson Hall
University of Nebraska
Lincoln, NE 68588

A compendium of articles on the application of Piaget's models in college teaching. In addition to updating this compendium on a more or less regular basis, the ADAPT faculty conducts workshops on Developing College Students' Reasoning for college teachers, administrators, and instructional development personnel.

TEACHING THE COBOL SEQEUNCE

Robert T. Grauer, Ph.D.

University of Miami

This paper is extracted from a one-day workshop, "Teaching the COBOL Sequence", conducted at ISECON-83. A "student contract", or written statement of course requirements, is made the focal point of the CIS-II syllabus. The importance of early entry on the computer is stressed, and general course guidelines are presented. The COBOL curriculum is extended beyond the language, based on the premise that mere knowledge of COBOL does not yield a competent professional. A control language must be taught and pedagogic suggestions are offered. The relationship to an Assembler language should also be included in the COBOL curriculum.

Programming is an exact profession with well-defined and precise specifications. It follows, therefore, that programming classes should also have clear and unambiguous specifications. This is best achieved through presentation of a written student contract, on the first day of class. This document clearly details the course requirements including project specifications and grading policy.

The student contract specifies when projects are due, and the instructor is well advised to adhere rigidly to the indicated dates. Late projects ought not to be accepted, no matter how creative the excuse. Students should be forewarned that the system will go down the day before a project is due, that terminals will be hard to come by, etc. No matter. The instructor must be firm and consistent in insisting that projects be submitted promptly; "tough but fair" is a highly prized reputation.

The contract should also contain the date of examinations and the grading criteria. Opinions differ as to the relative weight given to projects versus examinations. The author believes that since the course emphasis is on programming per se, the grading policy should likewise reflect this effort. A typical load is five programming projects, two in-class tests, and a final. The five projects and two tests count as seven equal components in computing a grade. Students may exempt the final with an A, given an overall 90 average on the seven components, as well as a 90 average on the two exams. (The latter precludes exempting the individual with five 100's on the projects, and two 65's on the tests.) Students failing to exempt have eight equal components to their final grade. A substantial number of A's and B's will emerge as final grades. However, the atypical grade distribution is reasonable, given the significant student effort of 10-15 hours per week in the computer center.

EARLY ENTRY ON THE COMPUTER

After the course requirements and grading policy have been clearly explained via the student contract, the instructor proceeds with the business of teaching. In general, there are two conflicting approaches to the presentation of COBOL. One can devote several lectures to the abundant rules associated with individual statements, or one can begin immediately with a complete program, leaving the details for later. The latter is suggested for the following reasons.

First and foremost, the computer is a better teacher than we are. Putting ego aside, a student learns far more in the computer center than in the classroom. A delay of several weeks in putting students on the machine only increases their level of frustration, with the end result that students are thoroughly bored by the time they get there.

Second, programming is fun, especially for those doing it for the first time. Indeed, one of the joys of teaching is in providing students with their initial exposure to the computer. Education and enjoyment need not be mutually exclusive; quite the contrary. The best courses, and the best instructors, are those that truly interest the audience. Students want to program, so be accommodating. Early entry on the computer is perhaps the single most important aspect of a successful course.

Figure 1 is the basis for the author's first lecture in "COBOL-I". It is a complete COBOL

```
IDENTIFICATION DIVISION.
PROGRAM-ID.    FIRSTTRY.
AUTHOR.        CRAWFORD.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT CARD-FILE ASSIGN TO UT-S-SYSIN.
    SELECT PRINT-FILE ASSIGN TO UT-S-SYSOUT.
DATA DIVISION.
FILE SECTION.
FD  CARD-FILE
    LABEL RECORDS ARE OMITTED
    RECORD CONTAINS 80 CHARACTERS
    DATA RECORD IS CARD-IN.
01  CARD-IN.
    05  CARD-NAME            PICTURE IS A(25).
    05  CARD-CREDITS         PICTURE IS 9(3).
    05  CARD-MAJOR           PICTURE IS A(15).
    05  FILLER               PICTURE IS X(37).
FD  PRINT-FILE
    LABEL RECORDS ARE OMITTED
    RECORD CONTAINS 133 CHARACTERS
    DATA RECORD IS PRINT-LINE.
01  PRINT-LINE.
    05  FILLER               PICTURE IS X(8).
    05  PRINT-NAME           PICTURE IS X(25).
    05  FILLER               PICTURE IS X(100).
WORKING-STORAGE SECTION.
77  DATA-REMAINS-SWITCH      PICTURE IS X(2)   VALUE SPACES.
PROCEDURE DIVISION.
MAINLINE.
    OPEN INPUT CARD-FILE, OUTPUT PRINT-FILE.
    READ CARD-FILE,
        AT END MOVE 'NO' TO DATA-REMAINS-SWITCH.
    PERFORM PROCESS-CARDS
        UNTIL DATA-REMAINS-SWITCH = 'NO'.
    CLOSE CARD-FILE, PRINT-FILE.
    STOP RUN.

PROCESS-CARDS.
    IF CARD-CREDITS NOT < 110 AND CARD-MAJOR = 'ENGINEERING'
        MOVE SPACES TO PRINT-LINE
        MOVE CARD-NAME TO PRINT-NAME
        WRITE PRINT-LINE.
    READ CARD-FILE,
        AT END MOVE 'NO' TO DATA-REMAINS-SWITCH.
```

Figure 1 - The First COBOL Program

program with trivial logic requirements, namely to process a file of student records and select all engineering majors with at least 110 credits. The author has long debated the merits of this listing as an initial lecture with many excellent instructors who argue "too much too soon". However, the COBOL student has already taken the traditional "Introduction to Data Processing", and consequently has been exposed to programming logic, files, records, fields, etc. Figure 1 is intended only to provide a conceptual view of COBOL, not a detailed discussion of COBOL syntax. An overall appreciation for COBOL can be achieved by supplementing the listing with several well-chosen exercises; e.g., analysis of test data and/or modification of logic requirements. The absolutely critical assignment, however, is for students to enter the listing as is, and obtain their own computer output. (This in turn requires a second lecture on the use of a text editor.)

Students will be overwhelmed by the amount of material in the early presentation, but not to an impossible extent. The instructor must provide words of encouragement about the importance of obtaining output at the computer center. Invariably, students (at least the better ones intending to complete the course) return after a week, with output, and calm returns to the classroom. Many will have compilation and/or logic errors despite the fact they copied the listing "exactly". That is secondary, however, to the fact that they have completed what, for many, is the most difficult assignment of the year. They have located the computer center, endured the waiting line for a terminal, learned a text editor, entered their program, found their output, etc. A new level of understanding has been achieved, and a firm foundation established on which to build the rest of the course.

Realize, also, that not every student will return, and that withdrawals will begin to trickle in. Don't be discouraged, because you have not failed as an instructor; a withdrawal rate of 25% or higher is typical.

GENERAL COURSE GUIDELINES

Once the initial panic and excitement have subsided, the class returns to a more traditional format. The following suggestions are offered as aids in conducting the course:

- Teach from COBOL listings to as great an extent as possible.

- Maintain continuity with multiple versions of the same program.

- Encourage walkthroughs and demand class participation.

Above all, recognize that the computer is the best instructor of all. Use it, therefore, by basing class lectures on actual programs. Select a text with a large number of programs that were actually run, and if necessary, supplement it with additional examples of your own. Present multiple versions of the same program in the early going, e.g., a correct listing, a modified version with compilation errors, a cleanly compiled version with logic errors, a contrasting version with or without coding standards, etc.

Students will learn the most from their assignments. Projects should be selected, therefore, with the utmost care, and varied each semester. The CIS-II syllabus of the Model Curriculum (1) describes in detail areas for project selection. Although this paper does not contain suggested projects per se, a summary of contrasting approaches is presented:

- Projects may be completely independent of one another, set in different contexts, with different data, etc.

- Projects may be independent of one another, but representative of different programs within a single system. This technique allows common elements of the Data Division to be reused in several programs along with common data files.

- Projects may be dependent on one another, e.g., the output of program 1 is input to program 2. Although this approach may be the closest to reality, it all but destroys the late starter.

Regardless of the approach chosen, the instructor is well advised to provide a common data file of his own to simplify grading.

BEYOND COBOL

Despite the so-called shortage of qualified data processing personnel, graduates of four-year programs are finding it increasingly difficult to obtain employment. The problem can, in part, be attributed to the recession, but at least an equal amount is due to the perennial lack of experience of the new graduate. He or she is faced with a classic catch-22, no employment without previous experience, but experience is obtained only through previous employment.

The educational community is doing a disservice to its constituents, if after extracting untold thousands of tuition dollars, its graduates cannot find gainful employment. It seems rather obvious that a partial remedy would be for the university to provide some of the requisite experience, or failing that, at least sufficient coverage of necessary material. It is high time, therefore, that we cease the disparagement of control language, memory dumps, etc. as subjects unworthy of coverage in a four-year curriculum. Instead, we must recognize that the COBOL programmer does not exist in a vacuum, and that knowledge of COBOL, in and of itself, does not yield a competent professional. A COBOL programmer must be able to interact with the operating system, to debug independently, and to leave well written and well documented programs for those who follow. In short, there is a need for several skills beyond COBOL.

THE ROLE OF JCL

The author does not advocate separate courses for JCL, utilities, ABEND debugging, etc., but believes that the material should be integrated with the COBOL presentation when the need occurs. Consider, for example, the traditional sequential update, shown in Figure 2, which is a mainstay of every "COBOL-II" course. The class assignment is

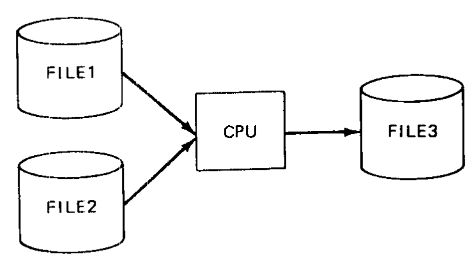
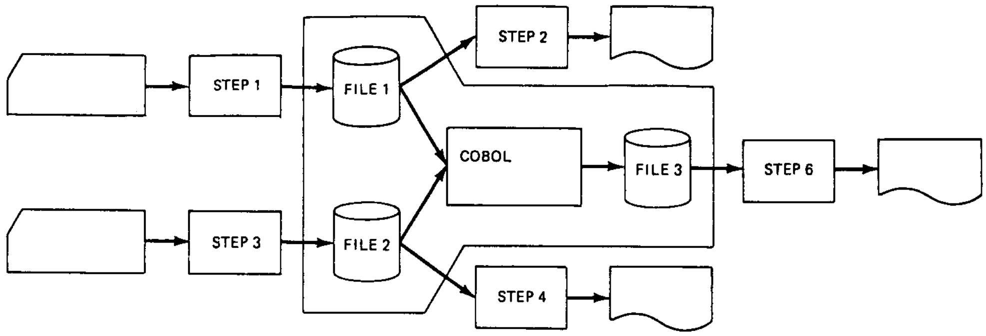


Figure 2 - The Sequential Update

to write the COBOL program to update FILE1 with transactions from FILE2, producing a new master, FILE3. However, in order to test the update program, one requires the availability of test data, i.e., FILE1 and FILE2. In addition, one has to print the contents of FILE3 after the update has taken place. The simple one-step assignment of Figure 2 becomes the six-step jobstream of Figure 3. What better way is there to motivate the need for utilities (e.g., IEBGENER under IBM's OS operating system). Adequate coverage of the IBM utility requires only an hour or so of class time, given a working knowledge of JCL.

Unfortunately, use of a control language,



Notes: Step 1 - creates first file
Step 2 - prints first file
Step 3 - creates second file
Step 4 - prints second file
Step 5 - COBOL assignment
Step 6 - prints merged file

Figure 3 - The Need for Utilities

especially IBM's OS JCL, retains a mystique which too often makes both student and instructor shy away. The typical graduate of a four-year curriculum has a COBOL capability far in excess of his or her command of the operating system. This individual is comfortable with the COBOL syntax for accessing an indexed file, for calling a subprogram, for using the SORT verb, etc., but relies exclusively on outside help for actual execution. "Just give me the control cards", or "Accept these on faith" are phrases heard far too often. The situation is not remedied by a separate course in JCL which is invariably dry and detached. It is solved by integrating the information with the COBOL presentation, and by presenting JCL for what it is, namely a means of communication between the programmer and the operating system.

Figures 4 and 5 illustrate the author's introduction to the subject. This exhibit, coupled with a statement that much of OS JCL revolves on only three statements (JOB, EXEC, and DD), is sufficient to allay preconceived notions about its difficulty. A two-hour presentation on syntax, keyword versus positional parameters, rules for continuation, etc. is sufficient to make students less uncomfortable, if not genuinely comfortable with the material. Proficiency comes with time and experience, as well as use of the information in the COBOL sequence.

THE NEED FOR ASSEMBLER

Although the COBOL programmer can, and does, function without knowledge of Assembler, even a superficial knowledge greatly enhances the capabilities of the individual in debugging and/or producing superior code. This is readily appreciated when students realize that the computer does not execute COBOL instructions per se, but rather the machine language instructions generated by the COBOL compiler. The DPMA Model Curriculum accepts this premise in its first elective, "CIS-8, Software and Hardware Concepts". The stated course description specifies "a survey of technical topics ... with emphasis on the relationships between hardward architecture, system software, and applications software".

Consider, for example, Figure 6, a simple COBOL program that illustrates both correct and incorrect ways of initializing group fields. The output values of FIRST-BINARY, FIRST-DISPLAY, and FIRST-PACKED are all one as expected. The value of SECOND-DISPLAY is also one, but the value of SECOND-BINARY is 3855 (that is not a misprint), and attempted execution of statement 47 (to calculate SECOND-PACKED) resulted in job termination and a subsequent dump. Why? Simply stated, the difficulties with SECOND-BINARY and SECOND-PACKED are caused by the group move of line 40. All group moves are treated as alphanumeric moves. Hence, the wrong kind of zeros are moved to SECOND-BINARY and SECOND-PACKED; in effect, these fields are not initialized to zero at all.

The preceding discussion is a technically correct, albeit unsatisfying, explanation of the problem. Nevertheless, it is all that is possible in a "pure" COBOL course which merely discusses the rules for padding and truncation associated with the MOVE statement. A deeper explanation is feasible only if the student has a basic understanding of Assembler fundamentals and internal data representation. The point is simply that COBOL and Assembler should not be taught as separate and distinct entities with no intersection, but that the advanced COBOL course must include reference to Assembler. In essence, the best COBOL programmers are those who know more than mere COBOL.

Programmer:	I have a job for you.
OS:	Tell me about it. In particular, I need to know the job's name, your name, how long the job will take, and whom to bill my services to.
Programmer:	The job's name is Example. My name is John Smith, it should take one minute and 30 seconds, and bill it to Programming.
OS:	Very well - What is the name of the program you will run?
Programmer:	Update.
OS:	How many files do you require, and what kind are they?
Programmer:	I have three files altogether, two input and one output. The input files are both on tape, and the output file is on disk.
OS:	You have to be more specific about your input files - Tell me on which tape volumes they are located, and the exact name of each file as it appears on the tape.
Programmer:	The first file is called OLD.MASTER.FILE and is the third file on the tape whose serial number is 111111. The second input file is called TRANSACT and is the first file on volume number 222222. Both have standard labels. Incidentally, my program knows these files simply as INPUT1 and INPUT2, respectively.
OS:	Tell me about your output file.
Programmer:	Call it NEW.MASTER.FILE and put it on any disk you want. I estimate that it will require 10 tracks. Keep it until the last day in 1985. My program knows this file simply as OUTPUT.
OS:	That should do it - Have a nice day.

Figure 4 - A Hypothetical Dialog

CONCLUSION

The following guidelines were suggested for presenting the COBOL curriculum:

- Use of a written student contract to unequivocally state the course requirements.
- Early entry on the computer, beginning with a complete program in the first lecture.
- Constant presentation of complete programs in class; also, multiple versions of the same program with compilation and/or logic errors, with and without standards, etc.
- Integration of JCL and utilities within the second COBOL course.
- Integration of COBOL and Assembler to facilitate debugging.

REFERENCE

1. DPMA Model Curriculum For Undergraduate Computer Information Systems Education, David R. Adams and Thomas A. Athey, Editors, Educational Foundation, DPMA, Park Ridge, IL.

```
//EXAMPLE JOB (PROGRAMMING),'JOHN SMITH',TIME=(1,30)

//    EXEC PGM=UPDATE

//INPUT1 DD UNIT=TAPE,VOL=SER=111111,
//    DSN=OLD.MASTER.FILE,LABEL=(3,SL),
//    DISP=(OLD,KEEP)

//INPUT2 DD UNIT=TAPE,VOL=SER=222222,
//    DSN=TRANSACT,LABEL=(1,SL),
//    DISP=(OLD,KEEP)

//OUTPUT DD UNIT=DISK,DSN=NEW.MASTER.FILE,
//    LABEL=EXPDT=85365,DISP=(NEW,KEEP),
//    SPACE=(TRK,10)

//
```

Figure 5 - JCL as a Means of Communication

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID.   GROUPMV.
       AUTHOR.       R GRAUER.


       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER.       IBM-370.
       OBJECT-COMPUTER.       IBM-370.


       DATA DIVISION.
       WORKING-STORAGE SECTION.
       77  FILLER                  PIC X(14)  VALUE 'WS BEGINS HERE'.
       01  FIRST-GROUP-FIELD.
           05  FIRST-DISPLAY       PIC 9(4).
           05  FIRST-BINARY        PIC 9(4)     COMP.
           05  FIRST-PACKED        PIC 9(4)     COMP-3.

       01  SECOND-GROUP-FIELD.
           05  SECOND-DISPLAY      PIC 9(4).
           05  SECOND-BINARY       PIC 9(4)     COMP.
           05  SECOND-PACKED       PIC 9(4)     COMP-3.

       PROCEDURE DIVISION.
      * CORRECT INITIALIZATION
           MOVE ZEROS TO FIRST-BINARY.
           ADD 1 TO FIRST-BINARY.
           DISPLAY 'FIRST-BINARY = ' FIRST-BINARY.

           MOVE ZEROS TO FIRST-DISPLAY.
           ADD 1 TO FIRST-DISPLAY.
           DISPLAY 'FIRST-DISPLAY = ' FIRST-DISPLAY.

           MOVE ZEROS TO FIRST-PACKED.
           ADD 1 TO FIRST-PACKED.
           DISPLAY 'FIRST-PACKED = ' FIRST-PACKED.

      *INCORRECT INITIALIZATION
           MOVE ZEROS TO SECOND-GROUP-FIELD.
           ADD 1 TO SECOND-BINARY.
           DISPLAY 'SECOND-BINARY = ' SECOND-BINARY.

           ADD 1 TO SECOND-DISPLAY.
           DISPLAY 'SECOND-DISPLAY = ' SECOND-DISPLAY.

           ADD 1 TO SECOND-PACKED.
           DISPLAY 'SECOND-PACKED = ' SECOND-PACKED.
           STOP RUN.
```

Figure 6 - The Need For Assembler

ON THE NECESSITY OF TEACHING APPLICATION DEVELOPMENT WITHOUT APPLICATION PROGRAMMERS

Perry Sanders

Indiana University Northwest
Gary, Indiana

The features of application development without application programmers are briefly reviewed. The quantam leaps in improvement in hardware price-performance is contrasted with the relative stagnation of third generation software development price-performance. An important conclusion is that application development without application programmers is a rising trend and will become the dominant trend in the information systems industry. The need for immediate Computer Information Systems (CIS) faculty education in this area is advanced. A course in application development without application programmers is proposed for the CIS curriculum, and an outline proposal for such a course at Indiana University Northwest is presented. The recent DPMA[17] and ACM[18] undergraduate curriculums in (C)IS are criticized for lacking a proper fourth generation orientation and emphasis toward application development without application programmers. Finally, interest is expressed in working with various parties on concretely resolving this stated problem.

WHY APPLICATION DEVELOPMENT WITHOUT APPLICATION PROGRAMMERS?

What Is Application Development Without Application Programmers?

Application development without application programmers is application development and maintenance without second (assembler) or third (e.g., COBOL, PL/I) generation hand-coding. In place of assembler, COBOL, ADA et al. non-procedural languages (e.g., FOCUS[1], QBE[2], and RAMISII[3]), non-procedural/very high level procedural languages (e.g., NOMAD2[4], MANTIS[5]), and axiomatic, specification-generation languages (e.g., USE.IT[6]) are utilized. With a non-procedural language the user describes what is to be done. On the other hand, a user of a procedural language details how the solution is to be obtained.

Some of these types of languages are suitable for use by end-users; that is, easy to learn a powerful subset in one-two days, easy to use, and generally non-procedural (e.g., FOCUS[1], NOMAD2[4], QBE[2], RAMISII[3]); while others tend to be more suitable for systems analysts (e.g., USE.IT[6]) or programmer-analysts (e.g., MANTIS[5]). Languages utilized by end-users are also used by Information Systems professionals, often with greater scope and efficiency. Such languages range from sophisticated database query languages (e.g., QBE[2]) to varying degrees of generalized application generators (e.g., RAMISII[3], FOCUS[1], USE.IT[6]) to report generators (e.g., NOMAD2[4]) to very high level programming (e.g., MANTIS[5], NOMAD2[4]). Languages in the categories mentioned above--there are many in each, except for USE.IT[6], and perhaps NOMAD2[4]--are often referred to as fourth-generation languages. To be properly termed fourth-generation the users of such languages must obtain results in one-tenth of the time or less compared to hand-coding in COBOL.[7]

In many cases application development without application programmers is also application development without the traditional, voluminous requirements and specification definitions, and specifiction freezes. In place of these components of the formal or structured systems life cycle, prototyping is utilized. A prototype has six salient characteristics: 1) it is a functioning system; 2), it may become the actual production system; 3) it is used to test out assumptions; 4) it is built quickly; 5) it is relatively inexpensive to build (compared to COBOL); 6) building a prototype is an iterative process.[8] The software for developing prototypes is entirely generative (e.g., RAMISII[3]) or basically generative (e.g., DMS[9]).

Application development without application programmers is also the Information Center, an organizational form for end-user computing. Information centers developed within IBM, and are promoted by IBM[10] as a highly cost-effective means of reducing applications backlog and satisfying end users' needs if established, managed, staffed, and developed properly.[10] The Conference keynote speaker, Dr. Gordon B. Davis, has warned against the dangers involved in user-developed decision-support systems (a subset of end-user computing) and proposed remedial measures.[11] The Information Center is an organizational vehicle for achieving application development by end-users, and end-users and systems analysts, all without application programmers.

James Martin advocates Information Centers with a broad scope of activity for the Information Center staff.[12] In addition, for medium and large-sized enterprises, Martin has recently proposed splitting traditional Data Processing development into an Advanced Development Center and a Traditional Development Center. The former would utilize Higher Order Software, Inc. methodology-USE.IT[6] and non-procedural languages, while the latter would continue with second and third generation programming languages. The Information, Advanced Development, and Traditional Development Centers would compete against each other and report to the Director of Management Information Systems. The goal would be to progressively minimize and eventually end traditional programming.[13]

Finally, application development without application programmers is also the utilization of application packages, preferably parameterized application packages. Developed by software houses, some of these packages are becoming increasingly generalized within a specific area (e.g., Accounts Payable). Excellent parameterization of application packages can enable users to customize implementation, within limits. Thus the trap of hand modification (e.g., in assembler or COBOL) is avoided.

Application development without application programmers is a rising trend. It is many things, and goes by many different names. It requires good management, good database administration and design, good communication skills, etc. The next section discusses the underlying economics behind this rising trend.

Data Processing Economics of Application Development Without Application Programmers

During the decade of the 1970's, computing capacity increased by a factor of 10 with no increase in cost. On the other hand, structured analysis, design, and programming probably improved productivity by a factor of 1-2. However, such improvement was offset partially by the increased cost of analysts and programmers.

The cost of a MIP (Million Instructions per Second) of computing capacity in the mid'70's was about $40,000 per month compared to about $2,000 per month for a software specialist, a cost ratio of 20:1.[14] By 1979 it has been stated that the cost of renting 300K instructions per second of computing capacity was less than the cost of a Data Processing professional for a comparable period of time.[15] According to Dr. Lewis M. Branscomb, Chief Scientist of IBM, "By the mid-eighties, for example, it is estimated that renting one million instructions per second of computing capability will cost less than hiring a professional programmer for the same period of time."[14]

In the next seven years, by 1990, the price-performance ratio of computing capacity should yield an improvement of another order of magnitude.

The hardware-software economics already favor application development without application programmers. In most instances this is true even if such development is less hardware efficient.

To fully exploit the existing and future hardware advances rather than drown in the mounting tide of applications backlog, stated and hidden, there is only one solution: automate the automaters. Of course, this can only occur progressively and relatively. Yet it must. It will.

INTRODUCING APPLICATION DEVELOPMENT WITHOUT APPLICATION PROGRAMMERS INTO THE CIS CURRICULUM

CIS Faculty Education In The Area Of Application Development Without Application Programmers

This section starts with the conclusion that since application development without application programmers is in fact a rising trend, and will probably be the dominant trend by 1990, it is necessary to begin changing the CIS curriculum in this direction. For obvious reasons, a critical factor in accomplishing this is CIS faculty education.

Carrying out CIS faculty education in the area of application development without application programmers is especially difficult at this time. CIS faculty are not in the same position as practicing Information Systems professionals. IS installations are being pushed in the direction of application development without application programmers by the factors described earlier, in addition to the developing economic depression[16] which is restricting hiring options. On the other hand, CIS faculty face surging undergraduate enrollment demands semester after semester. This is true even if overall enrollment at particular institutions declines. More sections of structured COBOL, etc. are added. More full-time and part-time faculty are added, if they can be found. The students keep coming! The demands on CIS faculty are enormous.

The ambush ahead is compelling though. Within a few years the information systems industry will swing hard towards application development without application programmers. Simultaneously, the growing number of CIS academic programs, and the growing CIS departments will be turning out a significant number of entry-level programmer-analysts skilled in COBOL application development. Pow! Overproduction. Those that consider this to be preposterous would do well to keep in mind that nothing in the past continues forever.

Ways must be found to facilitate CIS faculty education in the area of application development without application programmers.

This recognition was a primary impetus behind my decision to offer to conduct a workshop at ISECON83 in the area of: Teaching Application Development Without Application Programmers. Besides specific concretes of application development without application programmers, some concrete suggestions for future CIS faculty education in this area will be made at the above Conference workshop.

Find Ways to Introduce Application Development Without Application Programmers

This should be done in numerous courses. For example, in the first course in the CIS curriculum the economics and prominent features of applications development without application programmers could be introduced. With the first course in database systems, use of database query languages could become an essential component. The first course in analysis and design could include a component on prototyping. And on and on. The inclusion of such components is important, but it should be accompanied by a highly significant, immediate addition to the CIS curriculum.

Implement a Course in Application Development Without Application Programmers

Such a course is needed if for no other reason than a CIS faculty statement that application development without application programmers is, at least, a vital component of the CIS curriculum. Such a course would permit application development without application programmers to be dealt with in a focused manner over an extended period of time. The Seniors of 1983 need such a course!

At Indiana University Northwest, in the Fall of 1982, the Department of Data Processing and Information Systems (DP & IS) proposed a new elective course: Application Development Without Application Programmers, D446. We proposed first teaching it in the Spring of 1984. At Indiana University new courses have to go through a lengthy approval process. A basic outline for this course runs along the following lines:

I. Problems with conventional development and maintenance
II. Where conventional development is necessary
III. Hardware economics and the economics of software development
IV. Application development without application programming
   A. Preprogrammed application packages
   B. Application generation by systems analysts
      1. Prototyping
      2. Very high level procedural languages
      3. Non-procedural languages
   C. Application development by end-users
      1. Database languages
      2. Very high level languages
      3. Application generation
V. Information Centers
VI. Resistance to change
VII. Future prospects in industry

NOTE: Students will complete practical work.

THE NEED TO CONTINUE WORK ON A CIS CURRICULUM FOR THE FOURTH-GENERATION

There are many important things about both the DPMA Model Curriculum for Undergraduate Computer Information Systems Education[17] and the ACM Information Systems Curriculum Recommendations for the 80's: Undergraduate and Graduate Programs.[18] Certainly, there are too many important things to elaborate on here. However, a critical problem with both undergraduate curriculum recommendations is that they lack a proper orientation and emphasis toward application development without application programmers. Both are rooted in the systems life cycle of the third generation. The DPMA Model Curriculum emphasizes third generation application programming much more than its ACM counterpart. Although many courses and ideas are right, e.g., CIS 10 Decision-Support Systems,[17] how is it possible for these curriculum recommendations to have a fourth-generation content without all the components of application development without application programming? The ordering, scope, and depth of the components are important factors too.

At Indiana University Northwest the DP & IS faculty are beginning to carry out global fourth-generation curriculum development work. Presently, our Bachelor of Science degree program in DP & IS (with a Business Administration concentration) more closely resembles the DPMA Model Curriculum than the new ACM recommendations. We hope to complete our basic curriculum development work in the Summer of 1983. Certainly, the author would like to work with interested parties on undergraduate curriculum development for the fourth-generation.

REFERENCES

1. FOCUS is a product of Information Builders, Inc., N.Y., N.Y. FOCUS will be demonstrated at the workshop mentioned in this article.
2. QUERY-BY-EXAMPLE is a product of IBM.
3. RAMIS II is a product of Mathematica, Inc., Princeton, N.J.
4. NOMAD 2 is a product of National CSS, Wilton, CT.
5. MANTIS is a product of Cincom Systems, Cincinnati, OH.
6. USE.IT is a product of Higher Order Software, Inc., Cambridge, MA. USE.IT will be demonstrated at the workshop mentioned in this article.
7. Martin, James, Application Development Without Programmers, Prentice-Hall, Inc., 1982, p. 28.
8. "Developing Systems by Prototyping," EDP Analyzer, Sept. 1981, p. 6.
9. Development Management System is a product of IBM.
10. Hammond, L.W., "Management considerations for an Information Center," IBM Systems Journal, Vol. 21, No. 2, pp. 131-161.
11. Davis, Gordon B., Caution; User-Developed Decision-Support Systems Can Be Dangerous to Your Organization, MISRC, Minneapolis, MN.
12. Martin, James, "Information Center Management" and "Information Centers in Practice," Application Development Without Programmers, Prentice-Hall, Inc., 1982, pp. 298-330.
13. Martin, James, "The Changing Development Life Cycle," Program Design Which is Provably Correct--A Quantum Leap in Software, Savant Institute, Cornforth, England, Dec. 1982, pp. 339-357.
14. Branscomb, Lewis M., "Bringing Computing to People: The Broadening Challenge," Computer, July, 1982, p. 71.

15. Martin, James, Application Development Without Programmers, Prentice-Hall, Inc., 1982, p. 3.
16. Forrester, Jay W., "Economic Decline Natural Result of 50-Year Cycle," Information Systems News, Jan. 10, 1983, p. 42.
17. DPMA Education Foundation Committee on Curriculum Development, DPMA Education Foundation, 1981.
18. ACM Curriculum Committee on Information Systems, Communications of the ACM, Nov., 1982, Vol. 25, No. 11, pp. 781-805.

ISECON 83

Second Annual

Information Systems Education Conference

Building the Human Resource for Quality Computer Information Systems

MARCH 21-23, 1983 **CHICAGO, ILLINOIS**

Conference Highlights

- Improving the Process of Technology Transfer Between University and Industry
- Cooperative Education Programs in Information Systems
- Business and Government Needs for Academically Trained Information System Developers
- Comparative Analysis of Information Systems Curriculum
- Vignettes of Information Systems Teaching Techniques
- The DPMA Model Curriculum: An Update
- Dean's Eye View of MIS

Four Full-Day Practical Workshops

- Teaching Learning By Design
- Teaching The COBOL Sequence
- Teaching Information Systems Design
- Teaching Application Development Without Application Programmers

Keynote Address
Evolution of Information Systems as an Academic Discipline
Dr. Gordon B. Davis

Closing Address
Lessons of the Conference
Dr. Harlan D. Mills

Sponsored by
DATA PROCESSING MANAGEMENT ASSOCIATION
EDUCATION FOUNDATION

ISECON83 Second Annual

Information Systems Education Conference

March 21-23, 1983
McCormick Inn
Chicago, Illinois

Conference Objective

Reliable and relevant data processing can exert enormous leverage on the nation's productivity and economic health. Such data processing is critically dependent on the knowledge and technical skills of the workforce that develops and maintains the computer information systems of the nation's enterprises.

The objective of the annual ISECON conference is to accelerate the development and delivery of education to the present and future workforce as rapidly as possible. ISECON is intended to play a role in preparing the nation and its DP managements for the future with a better educated and more effective workforce.

The theme of ISECON83 is building the human resource for quality computer information systems. This theme was chosen in recognition of the remarkable human potential and opportunity for developing information systems at a new level of quality, reliability, and responsiveness. Effective education is crucial for tapping this human potential, both in the transfer of technical skills, and in the creation of responsible, but challenging, expectations of excellence in information system development.

Program Committee

Program Chairman
M.H. Schwartz
General Manager
Software Quality Service, Inc.

Publications Chairman
Richard C. Linger
Federal Systems Division
IBM Corporation

Members

Dr. Gerald Hoffman
Manager, Operations Research
Standard Oil (Indiana)

Dr. Herman Hoplin
Associate Professor of MIS
Syracuse University

F. Walter Jenison, Jr.
Chief, Systems Development Service Branch
Tennessee Valley Authority

Dr. H. Albert Napier
Associate Professor MIS
Director of Information Systems Research Center
University of Houston

Herbert Safford
Staff Administrator
GTE Data Services, Inc.

Celesta Weise
Manager, Dallas Advanced Education Center
IBM Corporation

Dr. James C. Wetherbe
Associate Professor & Director, MIS Research Center
University of Minnesota

About the Sponsor

Data Processing Management Association Education Foundation

The DPMA Education Foundation, as an independent body representing practitioners, educators and researchers, seeks to provide leadership in identifying educational opportunities that will advance the information systems profession. An immediate Foundation objective is the dissemination of information among technical societies and the entire information systems community.

CONCURRENT PRE-CONFERENCE WORKSHOPS

Monday, March 21, 1983
9:00 AM - 5:00 PM

To ensure that all who attend a Workshop will have the opportunity to obtain the most from these practical, full-day learning opportunities, enrollment will be limited to no more than 100 in each. Please tell the Registrar which Workshop you wish to attend when you call or write to register.

Teaching Learning By Design

Dr. Daniela Weinberg
Dr. Gerald M. Weinberg
Weinberg & Weinberg

Learning is most efficient when poeple learn through direct interaction with their environment. This workshop demonstrates why, and shows how to create effective experiential learning programs.

Teaching the COBOL Sequence

Dr. Robert T. Grauer
Associate Professor
University of Miami

This workshop presents various techniques for teaching the 1-year COBOL sequence be it in industry or the university. Attendees are provided with abundant handouts which may be applied immediately in their respective classrooms. The seminar stresses that knowledge of COBOL alone does not guarantee a competent programmer. Suggestions are made for integrating peripheral topics with the COBOL presentation.

Teaching Information Systems Design

Dr. Nancy Martin
Professor, School of Information Technology
Dr. William McKeeman
Professor, School of Information Technology
Wang Institute of Graduate Studies

After reviewing problems experienced by industry in the production of information systems, we will show, using two specific applications, how the use of proper notation, design principles, verification and validation techniques result in better software. Throughout the day the emphasis will be on methods for teaching techniques and principles.

Teaching Application Development Without Application Programmers

Professor Perry Sanders
Chairman and Associate Professor of Data Processing
Indiana University, Northwest

Fourth generation economics, software, and organization. Application generation by systems analysts and end users. Demonstration. Building provably consistent and logically correct systems without applications programming. Information centers. Resistance. Future prospects. The computer information systems curriculum and fourth generation reality.

Tuesday, March 22, 1983

AM

7:45 Registration

8:45 **Welcoming Remarks**
Terrence J. Boyer
President, DPMA Education Foundation
V.P. and Manager, Information Planning & Control
Mercantile Trust Company, N.A.

9:00 Keynote Address:
Evolution of Information Systems as an Academic Discipline
Dr. Gordon B. Davis
Honeywell Professor of Management Information Systems
University of Minnesota
- Academic antecedents of information systems
- Technology connection
- Necessity for information systems to be separate from computer science
- Intersecting domains of information systems and traditional academic discipline
- Signs of an information system discipline
- Central tendencies in information systems research
- The practitioner/academic interface

10:00 Refreshments

10:30 **Business and Government Needs for Academically Trained Information Systems Developers**
- Executives from government and industry express their needs
- What are the roles of vocational programs?
- What are the roles of undergraduate programs?
- What are the roles of graduate programs?
- What should the differences be between computer science and information systems programming?

Moderator: **Dr. James Wetherbe**
Associate Professor and Director
MIS Research Center
University of Minnesota

Panelists: **George Perry**
Vice President, Information Systems
Investors Diversified Services, Inc.
Dr. Gerald Hoffman
Manager, Operations Research
Standard Oil (Indiana)
F. Walter Jenison, Jr.
Chief, Systems Development Service Branch
Tennessee Valley Authority

PM

12:00 Lunch break

1:30 **The DPMA Model Curriculum: An Update**
Dr. Thomas I. M. Ho
Head, Department of Computer Technology
Purdue University
- Status of CIS programs
- Identification of prototype programs
- Self-assessment and evaluation of adopting institution
- Creating a cadre of model curriculum implementers
- Where do we go from here?

2:30 **Comparative Analysis of Information Systems Curriculum**
Dr. William W. Cotterman
Professor of Information Systems
Georgia State University
- What is a standard curriculum really?
- What is the difference between the ACM and DPMA curricula?
- Which one should I choose?
- How should I use a standard curriculum?
- Are there any general principles of information systems education?

3:30 Refreshments

4:00 **Improving the Process of Technology Transfer between University and Industry**
Dr. Edgar Sibley
Executive Vice President, Alpha Omega Group
Former Chairman, Department of Information Systems Management, University of Maryland
- Under what conditions is the transfer of information systems technology a problem?
- Is this specific to the data processing profession?
- Where this is a problem, who is to blame—industry, academia, or both?
- How can the process be improved?
- How can the professional societies help?

5:30-7:30 **CONFERENCE RECEPTION**

Wednesday, March 23, 1983

AM

8:30 **Cooperative Education Programs in Information Systems**
- Teaching strategies and novel approaches
- Work related experiences in higher education
- Intern programs
- Project courses

Moderator: **Celesta Weise**
Manager, Dallas Advanced Education Center
IBM Corporation

Panelists: **Jack N. O'Day**
Data Processing Instructor
Cooperative Program Coordinator
Delaware Technical Community College
Dr. Gustave Rath
Professor, Industrial Engineering and Management Sciences
Northwestern University

9:30 Refreshments

10:00 **Vignettes of Information Systems Teaching Techniques**
- Analysis and design
- Data management
- Decision support systems
- Human resources management

Moderator: **Herbert Safford**
Staff Administrator
GTE Data Services, Inc.

Panelists: **Mary E. Rasley, CDP**
Assistant Professor for D.P.
Lehigh County Community College
Dr. Herman Hoplin
Associate Professor of MIS
Syracuse University
Dr. Lois Graff
Associate Professor of Management Science
School of Government and Business Administration
George Washington University
Dr. Kathryn M. Bartol
Professor and Chairperson, Organizational Behavior and Industrial Relations Faculty
University of Maryland

PM

12:00 Lunch break

1:30 **Dean's Eye View of MIS**
Dr. Lawrence E. McKibbin
Dean, College of Business Administration
University of Oklahoma
President
American Assembly of Collegiate Schools of Business
- What can and should the university deliver in the MIS field?
- Supply of and demands for MIS faculty
- Where does MIS fit in the university structure?
- Faculty development and evaluation
- AACSB role in the university MIS program
- Cost benefit analyses of MIS programs

2:30 **Lessons of the Conference**
Dr. Harlan D. Mills
IBM Fellow
IBM Corporation
Professor of Computer Science
University of Maryland
- Significant points of the speakers
- Significant points of the panels
- Global summary of substance and implementation
- Implementing what we've learned

ISECON83

Second Annual Information Systems Education Conference

Building the Human Resource for Quality Information Systems

Chicago, Illinois
March 21- 23, 1983
McCormick Inn
23rd and the Lake • Chicago, IL 60616

Registration: Entire fee is payable prior to the Conference. Make check payable to **ISECON83.**

Conference Fees: (includes conference proceedings, workshop materials, reception and refreshments).

	DPMA Members	Non-Members
Conference Only	$125	$150
Workshop Only	$100	$125
Conference & Workshop	$200	$225

These *reduced fees* are made possible by a grant from the IBM Corporation.

Limited Registration: Registration for this Conference will be limited to first 500 paid. Registration for each Workshop will be limited to first 100 paid.

Registration Desk: The registration desk will be open Sunday, March 20, from 5 to 8 p.m., Monday from 7:45 to 9:00 a.m. and 5 to 8 p.m., and Tuesday from 7:45 to 8:45 a.m.

Accommodations:
These special academic rates have been arranged with the McCormick Inn for **Isecon83**.

Singles	$55.
Doubles	$65.
Suites	$188 to $700.

All rates are subject to city and state hotel/motel tax of 9.1%. Conference attendees may make their own arrangements to double-up. Special arrangements have been made for inexpensive lunches to be available at the Conference.

Changes to the program may be made without prior notice.

IT'S EASY TO REGISTER! Call
(301) 445-4400

AIRFARE SAVINGS
American Airlines is working with your travel coordinator, CGT, in saving you travel dollars.
- Save a minimum of 25% off full fare American Airlines coach fare
- Advice on how to qualify for lower air fares
- Car rental reservations.

Call **American Airlines** Toll Free: 1-800-433-1790 (In Texas, 1- 800-792-1160). Ask for Star File #5414.

------ *Please cut here and return entire form* ------

Please type or print

Name ______________________

Title ______________________

Phone (______) ______________________

Organization ______________________

Street ______________________

City ______________ State ________ Zip ________

☐ Conference

Please indicate by an (X) in the appropriate box below which of the Workshops you will be attending:

☐ Teaching Learning by Design
☐ Teaching the COBOL Sequence
☐ Teaching Information Systems Design
☐ Teaching Application Development without Application Programmers

☐ DPMA Member ☐ Non-member

Amount enclosed: $ __________

Return to:
USPDI
ISECON83
Department A B C D E
1805 Powder Mill Road
Silver Spring, MD 20903

NON-PROFIT ORGANIZATION
U.S. POSTAGE
PAID
PERMIT No. 3746
ROCKVILLE, MD

MAILROOM: Please RE-ROUTE if necessary. This important announcement, if not properly addressed, should go to the division responsible for **Information Systems Education/Training.**