CRITIQUE AND EVALUATION OF THE CAL POLY/DPMA MODEL CURRICULUM FOR COMPUTER INFORMATION SYSTEMS

William Mitchell
James Westfall

University of Evansville

ABSTRACT

The authors have been intimate observers of a significant movement within computer education. This paper presents a history of this curriculum project and an assessment of its future influence. The forces which mandate the focus of attention in the area of data processing education are identified and the nature of the response evoked from this project is analyzed. The paper reveals the need for a better understanding of curriculum development enterprises, and the necessity to promote greater cooperation both within the academic community and within the computer industry to insure that useful curriculum materials will emerge.

INTRODUCTION

The authors have been observers and participants in the process of constructing a new four-year curriculum in Business Data Processing. In addition, Dr. Mitchell is a student of the curriculum design process (9), a designer of an applied computing science program (10), an observer of the ACM curriculum committees (attending at least one session a year of each of the following committee meetings over the past 5 years: CAJC, ESSS, Accreditation, and Computer Science), and an active member of DPMA (currently chapter president). Mr. Westfall is a practicing computer professional (US Army, retired, contract programmer, consultant, OEM), who has begun a teaching career and is also a DPMA member. Both authors are "outsiders" with respect to the ACM establishment in curriculum development and with respect to the executive committee and the DPMA Education Foundation which governs the DPMA effort. They thus present as qualifications for the role of critic broad first-hand experience, relevant personal expertise, sympathetic interest and objective non-involvement.

HISTORY OF A MOVEMENT

The Cal Poly/DPMA model curriculum grows out of the personal inspiration and energy of Dr. Thomas H. Athey, Chairman of the Information Systems Department of the School of Business Administration, California State Polytechnic University at Pomona. Dr. Athey returned from the CSC conference in Detroit in 1978 with the conviction that no one else was better qualified to lead a development effort for a "practical" information systems curriculum. His was already the largest such department in the country and certainly one of the most progressive in offering a curriculum which met the needs of the burgeoning California data processing community. Dr. Athey proceeded to conduct a series of conferences which he has described elsewhere in greater detail than is summarized below (3).

In 1979 Cal Poly sponsored a national curriculum development workshop which drew principally from the West coast. At the workshop Dr. Athey presented the thesis that computer curricula should be classified either Computer Engineering (hardware), Computer Science (executive software), or Busines Data Processing/Information Systems (end user applications). From this view, reportedly adopted by the conference attendees, the work of ACM and IEEE in developing computer curriculum "neither apply to nor aid educators developing business information systems curricula (1)."

The conference participants then delineated the characteristics of the field of business information systems by analyzing the skills required to function in the commercial computer environment. Broad objectives of a new curriculum were outlined and the attendees enthusiastically agreed to support a second conference at which "specific course sequences will be proposed and examined. Course topics will include systems design and development, programming, small business computers and management information systems (1)." A committee composed of Athey, Dr. Jerry Wagner of Cal Poly, Don Medley of

Moorpark Community College and Tom Murphy of Golden Gate University was named to design a model.

One of the participants at a local follow-up meeting in June 1979 was Mr. Don Price, Dean of Computer Services at Sierra College and the new president of the DPMA Education Foundation. He was impressed with the curriculum's perspective and agreed to lend the Foundation's support.

The Second Annual Business Information Systems Curricula Development Workshop was held January 24 and 25, 1980, co-sponsored by Cal Poly and the DPMA Education Foundation. Primarily as a result of a half-page article in COMPUTERWORLD on December 10, the conference attracted 28 attendees from east of the Rocky Mountains. The 106 participants included 32 faculty from schools of business, 10 faculty from applied computer science departments, 26 community college instructors, 11 business and industry representatives, 10 publishing representatives and 6 high school teachers. The COMPUTERWORLD article quoted the organizers as seeking "To decide what standards undergraduate business DP programs should meet in preparing their students for jobs as systems analysts and programmers. To establish the business information systems field as a formal educational discipline, equal in standing to computer science instruction (5)."

The conference was to act as a forum for an exchange of ideas concerning model curricula, introductory DP course content, systems concepts and the development of educational materials for information systems departments. Although the unveiling of a new curricula was planned for NCC 1981, the Second Annual Workshop was seen "not as the climax of their curriculum development project, but as its starting point (5)."

In the opening session the conference participants were presented the model and rationale which had been developed by the curriculum committee (see figures 1 and 2). For the remainder of the conference the attendees listened to panel presentations (see figure 3), broke into groups for discussion, and listened to summaries of the conclusions of the various discussions.

The curriculum committee had felt "that the model curriculum should not define all courses that are taught in an information systems department but rather they should define a set of core courses which all schools should teach in a rather consistent fashion (2)." The conferees discussed no courses beyond the core, and even those courses were too many to debate in the four hours allotted to the groups. The most frequent discussion topics in the group discussions were the feasibility of a 10 course core and the ability of various institutions to teach the specific topics sketched for each course.

At the close of the conference it was announced that the next step would be a series of regional conferences which would gather input and reaction from each section of the country. Several participants expressed interest in organizing such meetings, but no schedule was suggested.

GENERAL OBJECTIVE: nationwide educational standards for BIS in 1980s.

SPECIFIC OBJECTIVES: model curriculum for applications programmer/analyst.
educational background to be project leader.
DPMA equivalent to ACM Computer Science 1968.
target market is community colleges and universities.

PLANNING ASSUMPTIONS: focus on information systems courses appropriate for 1982-87, which would be a required core for all semester calendar IS programs and would permit a smooth transition from community colleges to universities.

GENERAL TEACHING CONCEPTS: emphasizing integration of methodologies and "non-IS" skills in IS courses.

TECHNICAL ENVIRONMENTAL CONSIDERATIONS: including emphasis on terminal, on-line data base, distributed data processing context in which EDP auditing, make or buy decisions, and mini-micro computers would be increasingly important.

BACKGROUND NEEDED TO FUNCTION AS P/A: including ability to interact with users in a business organization, proficiency in team-oriented planning, anlysis, design and implementation activities as well as in-depth knowledge of a high level language and of the system and software development life cycles.

FIGURE 1. CURRICULUM RATIONALE

After the conference adjourned, an informal discussion among a dozen remaining attendees centered on the realities involved in meeting the NCC 1981 deadline and the potential affect of DPMA endorsement. No mechanism for that endorsement was identified. Dr. Athey expressed the conviction that an absolutely standard core offering was the only way of insuring the quality of the curriculum product. Consistent quality would earn recognition and exclusive patronage of the business community. DPMA's endorsement was to amount to an admonition to its members to hire only graduates of schools implementing the DPMA curriculum. A new executive committee was later created to oversee the continuing development of the DPMA curriculum (five of the six had been speakers at the conference: Athey, Price, Adams, Wagner and Stallard).

In March the participants and others who had written to express interest in the curriculum, each DPMA chapter, and the DPMA International officers were mailed a questionnaire and a copy of the model curriculum. This package included virtually the same rationale summarized in figure 1 and for each of the 10 course titles of figure 2 there was a one-page course content outline and some suggested assignments.

Lower division courses transferrable from the community college

INTRODUCTION TO COMPUTER BASED SYSTEMS general education course covering computer concepts, programming concepts, and uses and impact on society.

APPLICATIONS PROGRAM DEVELOPMENT I covering structured programming style, algorithm design and testing, building block concepts, batch--standalone.

APPLICATIONS PROGRAM DEVELOPMENT II covering dynamic information structures, testing and maintenance, interactive programs, and program module links.

STRUCTURED SYSTEMS ANALYSIS AND DOCUMENTATION covering documentation of current physical system, derivation of current logical system, establishing the system input and output, and data design.

Upper division courses offered only at universities

DATA BASE PROGRAM DEVELOPMENT covering derivation of the new physical system from the logical system, systems of programs, software development tools, and an actual DBMS.

STRUCTURED SYSTEMS DESIGN AND EVALUATION covering derivation of a new logical design, good design principles, constraints, and cost/benefit evaluation.

SYSTEMS PLANNING AND PROJECT MANAGEMENT covering determination of system problems, plans for solution development, hardware/software evaluation, and communication of ideas.

APPLIED SOFTWARE DEVELOPMENT PROJECT covering data base, teams, project management, walk thrus, modular testing, and presentations.

DISTRIBUTED DATA PROCESSING SYSTEMS covering data communication, distributed data bases, distributed systems applications and system alternative selection.

FUTURE INFORMATION SYSTEMS TRENDS covering MIS, office of the future, EDP auditing, privacy, DP law, and career aspects.

FIGURE 2. REQUIRED CORE COURSES

Out of about 500 surveys, 175 were returned. Seventy-eight of these were from educators, half at the university level. Ninety-seven were from industry, 60% being DP managers or vice presidents, 9% being programmers and 14% being analysts. Thirty-three respondents held memberships in ACM and 117 were members of DPMA.

"What Should Be Covered In A Programming Sequence?"
panelists: Tom Cashman, author
John Reutter, III, VISA
Lewis Myers, Jr., Systems Analyst, Univ. Texas at Austin

"What Should The Introductory DP Course Cover?"
panelists: Jerry Wagner, Cal Poly
Robert Behling, Boise State Univ.
Steven Mandell, Author

"What Systems Concepts Should Be Taught?"
panelists: James Stallard, General Dynamics
Georgia Miller, Administrative Systems and Business Ed, Indiana University
David Adams, Arkansas State University

"The State of the Art--DP in Industry and College"
panelists: Don Medley, Moorpark Community College
Sylvia Twomey, Cal Poly
Donald Davidson, LaGuardia Community College, NY

FIGURE 3. CONFERENCE SESSION TOPICS

The questionnaire required a response on a seven point scale between total agreement and total disagreement concerning each section of the curriculum proposal. Strong agreement (6 or 7) was expressed by 65-80% of the respondents with each of the objectives and assumptions (figure 1), but this fell to less than 50% strong agreement concerning the overall composition of the core, the number of recommended core courses, the number of recommended lower division core courses, and the number of recommended transfer courses from community colleges. The strong agreement rating of the descriptions of the content of the 10 core courses varied between 40% and 60%, with the second programming course being the most weakly supported. Only 35% of the respondents agreed that exactly two high level languages should be taught in the core, but 75% agreed that COBOL was the first choice. Dr. Athey has discussed the implications of this survey elsewhere (4).

The first regional conference to expose the model curriculum to the masses was held in St. Louis on October 23, 1980. The meeting was publicized in the DPMA chapter literature in September, and was announced in COMPUTERWORLD on October 6. Some direct mailing of brochures was made, but apparently not using the same list used for the March questionnaire. A revised curriculum dated August 1980 had been circulated to the executive committee members and a third revision, dated October 1980 was distributed at the St. Louis conference.

The August version had as its general objective the definition of a Computer Information Systems program for schools of business. The planning assumptions included an emphasis that only the transfer curriculum of the community colleges was being described, and that all core courses at both community colleges and universities would be taught "in a similar manner." A new section was added to the general background of the programmer/analyst which was entitled "derived requirements:" requirements emerging from the conference and the survey results. COBOL is now specified as the language to be used in the programming courses and BASIC is to be used in the introductory course. However, it is now felt that no more than 10 CIS courses should be required in the entire major, and that only seven of these courses should be in the core. The seven courses were selected from the previously proposed ten courses, but their content was extended so that no concepts were lost from the original core.

The St. Louis meeting attracted 55 participants, seven of which had been at Cal Poly in January (four were on the executive committee). This time the makeup was 30% business faculty, 30% applied computing faculty, 9% junior college instructors, and 17% business and industry representatives (most from the St. Louis DPMA chapter). All but the executive committee members and a New Hampshire faculty member were from the Midwest.

The October revision was so new that it had not been seen by all of the executive committee members at the conference (it is included in the Appendix). It differs from the August version by separating and elaborating on both the project objectives for establishing the model curriculum and the content that the model is to pursue. The mention of business schools is dropped from the general objective. The introductory course is modified to considerably decrease the general education emphasis and instead to stress programming (a sharp reversal of a major emphasis of the January conference). Little else is changed except that a Small Computer Software Development course is included as the twelfth elective (this is the only response to what is identified in the rationale as an important trend in the 1980s). The October version has been distributed to over 1000 individuals who have expressed interest in the curriculum from every corner of the globe. The results of the questionnaire accompanying the curriculum were to be available in December and were to guide the development of detailed course outlines and references by subcommittees chaired by recognized experts from each content area. The entire package will go to print in February 1981 in order to be available in quantity for distribution at NCC in May. In the meantime two regional conferences in January are planned for Dallas and Miami which will assess last-minute reactions. Two other meetings have been publicized for March in Toronto and Washington, D. C.

A variety of suggestions were made for changes in the present model at the St. Louis conference, the most concern being expressed about the content and emphasis of the introductory course, the role the curriculum plays within the community college, and the acceptabilty of the present model to AACSB accredited schools of business. One discussion group recommended replacing the core course on data base with an Applications Environment course surveying communications technology, distributive processing techniques, word processing and data base concepts in the spirit of the original CIS-10 course, Future Trends. Strong disagreement was voiced on the use of COBOL and BASIC as the only languages permitted in the core. A description of the conference and its relation to the curriculum effort appeared in COMPUTERWORLD on November 3, 1980 (8).

The regional conferences, if run as at St. Louis, are intended to fine tune rather than reorganize the model. The initial session by Dr. Athey made clear that the general and specific project objectives are no longer open to discussion. Once the curriculum model was distributed the participants were divided into discussion groups to dwell first on the topic "Philosophy for Model Curriculum" and later with the topic "Components of Model Curriculum." Actually, each designated group leader was provided with a list of questions for which they were to extract the group's response. The questions were essentially those which appear on the latest survey form which was distributed at the end of the conference. Those questions not resolved in the first discussion session were dogmatically pursued in the second session.

The October version of the model curriculum was presented by Dr. Athey on October 27 at a session of the DPMA International Convention, but the session was poorly attended and created little interest. No motion to endorse the curriculum was presented to the delegates.

THE MODEL'S OBJECTIVES

The Cal Poly/DPMA model curriculum aims to meet the need for establishing some standards for the training of business applications programmers. In an era when we are beset with new technologies and a plethora of structured methodologies, as well as intense economic pressure to improve productivity, the model intends to inject order and to insure a consistency of technical preparatin in the context of a business school environment. The present content emphasis is welcomed by large, sophisticaed data processing installations even though its embracing of "modern" methodologies may make it suspect in smaller shops. The curriculum identifies a variety of topics which are not addressed in any of the ACM or IEEE recommendations for four-year programs. The programmer/analyst it seeks to educate is neither a compiler writer nor an operations researcher. He is "people-oriented" rather than quantitatively or technologically skilled.

The ACM recommendations for MIS curricula (6) at the undergraduate level were dismissed by the executive committee because they were not pragmatically oriented. The delay of the Information Systems Curriculum Committee in publishing a revision of the 1973 recommendatins has been taken as an implicit endorsement of a very theoretical view of the MIS discipline. The Cal Poly/DPMA curriculum is therefore offered to guide the data processing educators in four-year programs whose interests have been ignored by ACM.

THE MODEL'S AUDIENCE

The Cal Poly/DPMA model curriculum committee has consistently identified its target as the computer information systems programs of community colleges and universities, with secondary interest in high school programs, proprietary schools, and DP training organizations. The original consumers of the curriculum are to be schools of business, who ought to be able to capture a major share of the industry's demand for bachelor-level graduates because industry will prefer a business applications orientation to any other. The proposed model has, however, been plagued by AACSB regulations which Cal Poly does not attempt to observe. The non-accredited business schools in general do not have the faculty resources to universally implement the model, and many would hesitate to embrace a non-accreditable program at a time when competition for students promises to intensify. The recent truncation of the core appears to be in acquiescense to AACSB's requirements for a common core and great breadth in business school graduates, but it is in conflict with the widely perceived emphasis on specialization as appropriate in the training of a productive programmer.

The community colleges are suspicious of the model curriculum because they see themselves already commissioned to meet the data processing programmer requirements of their locality and feel they are capable of doing that without a lot of integrated business concepts. Those students seeking to transfer, which continue to be a minority, have articulated requirements which interface with the accredited business schools and with university computer science programs in their region. The community colleges view the Cal Poly/DPMA model curriculum as endorsing a great portion of their vocational programs, but they resent the authoritarian requirements for literal implementation because it interferes with established relationships and their freedom to meet community needs.

The most receptive audience for the model curriculum are applied computer science programs, such as CSDP at Washington University which sponsored the regional conference at St. Louis. These programs have for many years pursued the training of applications programmers, both in business and in science. They found out years ago, as did Cal Poly, what would sell, and now have little need for guidance. These existing programs will, along with Cal Poly, be able to say "we have the DPMA curriculum" (even if it is not literally true) and despite the fact that few of these programs are within schools of business. Other computer science programs which are adding applied emphases in increasing numbers will find the model useful even though they will not attempt to achieve the kind of integration with business subjects which is demanded.

THE MODEL'S CONTENT

The seven core courses currently in the Cal Poly/DPMA model curriculum are a compromise to try to make the curriculum acceptable to AACSB schools (in the March survey less than 6% of the respondents felt that a core of seven or fewer courses was appropriate). The introductory course was originally modeled on standard texts in the area, such as Steve Mandell's. While there are obviously advantages for text authors to advocate that such a course be part of the nonmajor's core, or even a university-wide requirement, it effectively reduces the CIS major to nine courses instead of ten.

The lower divsion two-course COBOL sequence is assumed to give entry-level competence in programming. The only other programming course is the Applied Software Development course taught in the senior year to students who may not have programmed for a year. It is a team project-based course which will permit students to satisfy their programming requirement with a modest module. The emphasis on a potpourri of management and design concepts (data base, CRT screens, module interfaces, etc.), leaves little actual time for coding and debugging (less than 20%).

The other course on the programming side of the core, Data Base Program Development, seems to have even less emphasis on COBOL programming. It may not even involve a data base management system. The indication that the student will write a software system which will implement a network model of a DBMS is clearly a mistake. The student will be exposed to the network model of a DBMS and may even use a DBMS in an applications environment (why the network model is chosen rather than IMS or a relational model is unexplained--as is how this choice can be mandated universally despite local conditions). It is clear from the suggested topic list that 80% of the course is devoted to presenting, organizing and illustrating new concepts, leaving at most 20% for exercising those concepts by generating working code. The emphasis of such code will obviously be on the DDL and DML of the DBMS, and therefore the focus on the host language will be minimal.

It seems far-fetched to assume that the proposed courses will insure a graduate who meets the entry level requirements of industry (needs no immediate skill development to be productive). Certainly industry would expect a senior-level programming course (in COBOL) in which the student would gain some non-trivial coding experience beyond the simplistic exercises appropriate in the lower division sequence. The student will have much to learn about the use of systems utilities, the use of file management facilitites, and the properties of a standard COBOL compiler. Too much of the time in the present core which was once intended to achieve the desirable goal of "learning one language well" is spent on managing the software design and development process--which is language free.

Similarly, the two systems courses which remain in the core are spread too thin and placed too low in the curriculum to accomplish their goals. Little background is present to suggest that the student wil have any basis for designing physical systems or for evaluating designs and making language choices (points 8 and 9 of CIS-5). This is but one evidence that the Structured Systems Design course will be a survey course--the student will have to accept recipes which he will have no basis to question and no opportunity to adequately test. The systems sequence is unable to prepare a level 3 (entry level) competence in any of its topic areas. The Applied Software Project course is presented as a capstone course, but while it reconsiders several topics from the Data Base course and the second programming course, it spends scarcely any time on topics considered in the systems courses. Its description suggests that the student might somehow have gained the facility to use either PASCAL or BASIC at a capstone level--despite the explicit requirement that the core use COBOL.

Since the elective courses are not necessarily related to the core courses, a CIS graduate might complete his major with the Office of the Future, RPG and DP Law. Such a student is clearly not prepared to make his way as a programmer, if indeed his smattering of this and that has prepared him for anything at all in the technical side of data processing. Given completion of the business support courses, his breadth of training would seem to recommend him as a DP manager, but his lack of technical expertise would compromise his leadership as a project director. It is evident that the ten-course major is not deep enough.

EVALUATION OF THE MODEL'S ROLE

A business organization which hires a computer science graduate rightly complains that his commitment to business is suspect, his knowledge of business functions may be non-existent, and he must start from scratch to learn COBOL or RPG. Such an employee must be retrained at considerable expense. But the business organization may be confident that the CS graduate has a rich experience making programs work, and knows how to derive algorithms which are both efficient and correct. He will have a complete knowledge of the computer system and its functions, and will be accustomed to dealing with high levels of complexity. Given this experience it is a safe bet that the CS graduate can master the logic of control breaks and sequential file updates. Although he must learn COBOL, he will already have encountered many of its features in one of his previous languages. He will take a dim view on regressing to a first generation programming language, but he will be a staunch advocate of structured code, preprocessors and other productive techniques. Moreover, the problems of analysis and design and of software development are no different in a scientific environment than they are in a commercial environment--10,000 line programs are written in modules no matter what language is used, and they are developed in teams and tested systematically.

The major adjustment which the CS graduate must make to the business environment involves learning about business. To date we find that the non-technical education of a highly skilled technician is a task more easily performed than that of developing a high level of technical skill in a manager. Once the investment is made, the business organization has a very competent programmer whose interest is in developing quality software, not in becoming a senior analyst or DP manager as quickly as possible. He will enjoy the work he was hired to perform--programming. Should he develop other interests and skills on the job, he might migrate to analysis, but for the interim the organization will not suffer from his inability to perform the task at hand. Therefore, we suggest that it will be more productive to orient computer science students with a business minor than it is to give business students a CIS minor.

The fallacy of the Cal Poly/DPMA model curriculum is that it professes to serve two masters. It suggests that it produces a deep and rigorous technical education for applications programmers and at the same time develops a breadth of business and political skills required for practicing managers and analysts. In trying to do too much it does too little. The model points like a shadow to the path to be trod, but cannot itself deliver enough substance to be useful. The presentations of the October version emphasize that the model is a changing, "living" document which will surely be improved. This argues that a non-model is better than nothing at all. DPMA will have no option to disagree. It has been committed in advance to publicizing a curriculum which has now been emasculated so that it will meet the irrelevant criteria decreed by AACSB.

DPMA would be better served by a model which had as its goal the best training of the applications programmer. The applications programmer is an industry job category well understood, but it is not restricted to business applications or to being educated in schools of business. The problems of software development which plague data processing and engineering applications equally also plague large systems and small systems alike. Applied programs in computer science departments are free from AACSB regulations and are therefore free to enforce both technical depth and content specialization in application areas, be it business or science. The only cost is a weakening of the integration of the software project into a particular environment. But there is no one "business environment." Rather, there is an endless variety of individual applications, each requiring unique combinations of expertise (the banking industry applications are unlike the insurance industry's, which are unlike manufacturing applications, which are unlike retail applications). Is the student who has implemented a hotel reservations system any better prepared to work on a bill-of-materials application than the student who has implemented a text editor?

Programs such as recently announced by Indiana University (generally recognized as one of the top ten business schools), which capitalizes on learning applications management and design within the business school while utilizing the software development expertise in computer science (11), make the best use of existing educational resources. They are our best hope of raising the number and quality of DP-oriented graduates. Clearly DPMA will waste its influence on a solution which has little chance of being accepted or implemented in the universities because it offers little basis for improvement.

At the root of the self-limitations of the Cal Poly/DPMA model is a "them vs. us" mentality which isolates the executive commitee from the rest of the computer education community. The categorization of ACM as Computer Science (interpreted executive software) is a stereotype. It ignores the Community and Junior College Subcommittee which has published a thorough recommendation on the goals of training two-year programmers which enjoys wide consensus and influence in community colleges. It ignores the on-going work of ACM's Information Systems Curriculum Committee whose chairman offered Dr. Athey assistance and cooperation, but was rebuffed. The assumption by the executive committee that their concerns for data processing were unique ignores considerable work in software engineering sponsored by IEEE. It ignores a growing interest in applicatons programming evidenced by both large and small computer science departments outside of schools of business. It ignores a wealth of experience in data processing education possessed by DPMA members in academic institutions throughout the country (no effort has been made through DPMA to involve these members--the Education Foundation has entrusted the entire effort to Cal Poly). Finally, the calculated exclusion of any representative of an accredited business school (even a pragmatically oriented program such as those of the University of Minnesota, UCLA or Miami of Ohio) from the executive committee and the committee's provincal flavor ignore the advantages of gaining the consensus of the user community and permitting them to participate meaningfully in the design of the system.

CONCLUSION

The computer field is poorly organized and poorly defined. The computer related areas already encompass a breadth of applications which rival the engineering disciplines, but they do not have the unifying basis in mathematics and science which engineering enjoys. The skills acquired by some high school students in programming and analysis rival the abilities of many graduate students who study computer applications. There is no question that in any academic area there is a need for basic research, for applied research, and for technical training. But in the amorphous regions of computing it is unquestionably difficult to select all of the concepts of an entire new curriculum and package them into courses which will be attractive to any broad audience, if for no other reason than that there are no broad audiences to address. Computer applications are cursed with a mind-numbing specificity which isolates even one payroll package from another. If we have been able to agree on some approaches to executive software development it is because these applications at least must execute identically in every environment in which they are applied. It is difficult to imagine any business application which can make that claim. Therefore, while we can admire the good intentions which have motivated the model curriculum, we must question the wisdom of the effort.

The four-year academic programs in computing are being shaped by the market pressures which are exerted through the desire to expand long existing two-year programs and vocational-technical curricula to meet the manpower requirements of the data processing industry. The four-year programs are also strongly influenced by the pressures of graduate programs to produce students who will meet the rising entrance requirements which have been necessitated by the expanding complexity and specialization of of graduate studies. These trends are both powerful and to a great degree they are antithetical. It is not easy to find good compromises which protect the interests of the undergraduate students.

We believe that the goal of creating a four-year curriculum model which addresses the needs of the business data processing community is valid. The distinctions in orientation between Computer Engineering, Computer Science and Computer Information Systems must be clarified and understood. However, we find three critical weakenesses in the Cal Poly/DPMA model which prevent it from being a practical vehicle for achieving this goal in the academic environment of the 1980s.

1. The ten course major is inadequate to prepare the graduate for entry level positions in data processing. The model curriculum graduate would be better prepared for the business environment than most current computer science graduates, but at the expense of acquiring an adequate command of those technical skills so desperately needed by industry. Instead of being naive to business, the model curriculum graduate will be naive to the nuts and bolts experience of developing a realistically complex piece of software and making it perform efficiently.

2. The insistance on placing this curriculum in a School of Business is a mistake. Even though much applications expertise for this curriculum is available in a School of Business, very few such schools currently have the necessary depth of technical expertise to teach the software related skills. Given today's market conditions, it will be impossible for this deficiency to be corrected before the end of the decade. On the other hand, adequate software and hardware expertise is available outside of Schools of Business within most universities if efforts can be made to utilize it.

3. It may well be premature to completely define a new discipline because its practitioners have not yet reached consensus on its content and goals (7). It is surely ill-advised to demand rigid adherence to the initial design when that design is premised on a narrow, parochial view of business applications. While the derivation of the curriculum was orchestrated to give the appearance of wide participation, in fact the preconceived opinions of a handful of people were carefully protected. Only the opinions of those predisposed to agree with draft documents were solicited, and the tight development schedule left no room for other than minor variations. After announcing a fifteen month development period at the January conference in order to make the NCC 81 presentation deadline, and knowing then that the printing deadline was two months earlier, the managers of the development effort waited nine months before giving any computer educators an opportunity to criticize the product. When at last it becomes somewhat available for discussion (the regional conferences are not well publicized and the academic community is specifically ignored) we judge it neither to reflect the needs of industry nor the expertise available in academia.

With the assumption that the issue is not closed even though the curriculum to be presented at NCC 81 is already frozen, we make the following recommendations:

1. The model presented at NCC 81 should be presented as a working document rather than a recommended DPMA CIS Curriculum (25,000 printed brochures notwithstanding). Wide dissemination of the proposed model will invite comment from the diverse audience it supposedly will serve, and those comments should be systematically collected by an independent committee constituted by DPMA through the Vice President for Education. This committee should be representative of the DPMA membership and of all the educational institutions which serve the data processing industry. The current executive committee's outstanding contribution may well be that it will have made a four-year data processing curriculum a respectable issue and provided sufficient visibility that the industry and the educational establishment will take it seriously and provide support and status to the new committee.

2. The next model curriculum should not presuppose that CIS curricula must be offered solely by schools of business. It should instead concern itself with the skills and attributes desired in the graduates, and should suggest reasonable ways to develop them. The next curriculum attempt should evidence significant effort to help define a discipline by molding a consensus of the data processing educators as to the packaging and pacing of the curriculum's content. The next curriculum attempt should forthrightly address the issues which differentiate vocational training from baccalaureate education.

3. The next curriculum committee should attempt not only to identify those capablilites desired in entry level professionals in the near future, but it should seek to identify those areas of computer research which promise to have significant impact upon business applications during the professional life of the graduate. If a new discipline is to be fostered, it must be fostered at the theoretical research level as well as the entry level.

The Cal Poly/DPMA model curriculum is a significant effort. We believe that most of what it proposes is valuable. We are concerned that because the current model is seriously flawed its significance will be discounted. We advise computer educators in all areas to understand the flaws and to appreciate the circumstances which contributed to them, and then to seriously consider the motivations and intentions which even the flaws cannot obscure.

REFERENCES

1. Athey, Thomas, and Gerald Wagner, "Separate but Real, Business DP is Different Than 'Computer Science'," DATA MANAGEMENT 17,8 (August 1979), 36-37, 41-2.

2. Athey, Thomas and Jerry Wagner, "Preliminary model curriculum for business systems emerges, addresses 'real world' needs," DATA MANAGEMENT 18,4 (April 1980), 42-5.

3. Athey, Thomas, "Establishing A Curriculum For Computer Information Systems," INTERFACE 2,3 (Fall 1980), 22-25.

4. Athey, Thomas, INTERFACE 2,4 (Winter 1980), in press.

5. Beeler, Jeffry, "January Workshop to Begin Tackling University Curriculum for Business DP," COMPUTERWORLD, December]0, 1979, 2.

6. Couger, J. Daniel, "Curriculum recommendations for undergrduate programs in information systems," COMMUNICATIONS OF ACM 16,12 (December 1973), 727-49.

7. ICCP News Release describing the conference Systems Analysis and Design: A Foundation for the 1980s, held September 8-10, 1980 at Georgia State Univeristy, chaired by William W. Cotterman. Proceedings to be published by Elsiver North Holland in 1981.

8. Jones, Ron, "DPMA Model Curriculum to Push Business," COMPUTERWORLD, November 3, 1980, 29-30.

9. Mitchell, William, THE DESIGN OF MATHEMATICS CURRICULA FOR THE SMALL COLLEGE, Ph. D. dissertation, George Peabody College, 1974.

10. Mitchell, William and Bruce Mabis, "Implementing a Computer Science Curriculum Merging Two Curriculum Models," SIGCSE BULLETIN, August 1978.

11. Wise, Deborah, "Mesh Computers, Business," COMPUTERWORLD, October 20,1980, 30.

APPENDIX: DPMA'S MODEL CURRICULUM PROJECT

Revised October. 1980

CURRICULUM PROJECT OBJECTIVES

GENERAL PROJECT OBJECTIVE

1. To develop a nationally recognized and accepted model curriculum for undergraduate Computer Information Systems education for the 1980s.

SPECIFIC PROJECT OBJECTIVES

1. To identify the purposes and define the objectives of a model undergraduate curriculum in CIS.
2. To develop a core of CIS courses that would be common to all undergraduate programs leading to baccalaureate degrees in CIS.
3. To develop a set of elective courses that would supplement this core and would provide educational experiences to satisfy individual institution and student needs.
4. To define the objectives and to develop detailed outlines for all of the CIS courses in the curriculum.
5. To identify the appropriate academic levels at which these courses should be offered and the instructional tracks that should be followed by students in the curriculum.
6. To provide smooth articulation of courses between two-year programs in data processing and four-year programs in CIS.
7. To provide quality assurances and standardized educational outcomes among all undergraduate CIS programs.
8. To gain national acceptance of this curriculum as the model for undergraduate CIS education.
9. To establish CIS as a viable, but separate, curricular field in computer education.

CURRICULUM OBJECTIVES

GENERAL CURRICULUM OBJECTIVE

1. To provide graduates with the knowledge, skills, and attitudes to function effectively as applications programmer/analysts and with the educational background and desire for life-long professional development.

SPECIFIC CURRICULUM OBJECTIVES

1. To provide understanding of the goals, functions, and operations of modern business organizations.
2. To provide understanding of the information needs and of the role of information systems in business organizations.

3. To provide the analytical and technical skills for identifying, studying, and solving information problems in business organizations.
4. To provide communications and human relations skills for effective interaction with organization members, especially with the users and developers of information systems.
5. To provide the knowhow and ability for effective management of information systems projects.
6. To provide the background for further study of information systems.
7. To instill a professional attitude and seriousness of purpose about CIS as a career field.

PLANNING ASSUMPTIONS

1. The primary target markets for the model curriculum project are the CIS programs at the community college and university level. Secondarily, the interests of high schools, proprietary schools and the training programs of commercial organizations are considered.
2. The intent of the model curriculum is not to define all courses for community colleges but rather concentrate on those courses which students would take who want to graduate with a Bachelor's degree in CIS.
3. The Planning Horizon is 1982-1987.
4. Courses are to be based on a 3 unit-16 week semester system.
5. Core courses will not constitute more than 70% of the required CIS courses. This allows colleges flexibility to handle special individual community needs and faculty interests.
6. The freshman and sophomore level core courses are to be taught in a similar manner in all community college and university CIS programs.
7. The junior and senior level core courses are to be taught in a similar manner in all university CIS programs.
8. It is not the function of this model curriculum to develop individuals who are completely trained to function as programmer/analysts on specific computers for applications in specific industries.

TECHNICAL CONSIDERATIONS

1. Field is moving rapidly toward terminal, on-line, disk-oriented, database computer systems.
2. The mini/micro computers will come to be equally important in data processing as the main frames.
3. Field is moving away from assembly languages to higher-level procedural languages and toward non-procedural languages.

4. Need for more use of program development tools, aids, and standard logic functions.

5. Trend toward more make-or-buy decisions concerning applications software.

6. Trend for programmer/analysts to need to develop more expertise in the "front end" of the problem-solving cycle.

7. In addition to the accounting functions, there is need to start moving up the EDP Stages-of-Growth to consider planning, control, and decision support systems.

8. Greater usage of data processing by large, medium, and small companies.

9. That distributed data processing will soon be a part of many computer-based information systems.

10. The need for EDP Auditing is becoming more prevalent.

11. There will be a gradual merging of data processing, word processing, and data communication.

GENERAL TEACHING CONCEPTS

1. Unfold from simplified, specific concepts to more complex and abstract principles.

2. To keep students' interest, programming must be taught in the beginning CIS courses.

3. Don't have students fighting language, concepts, computer and background examples all at the same time.

4. Teach programming and systems concepts as integrated parts of computerized problem solving, as opposed to being separate disciplines.

5. Need to teach highly integrated methodologies, as opposed to a potpourri of techniques.

6. Have the student first learn skills individually. Then they can go to applying skills in a team environment.

7. If required "non-CIS" skills are to be grasped by the student as important, then they need to be integrated into the CIS courses and shown to be a "natural" part of the process.

GENERAL BACKGROUND NEEDED TO FUNCTION AS A PROGRAMMER/ANALYST

1. Be acquainted with the full systems life cycle.

2. Be able to function effectively in the planning, systems analysis and design, and program design, development, testing, and maintenance stages.

3. Be able to program indepth in 1 major high-level language.

4. Be able to work with user in providing computer solutions to business problems.

5. Be able to communicate ideas and project results orally and in written form.

6. Be able to function individually or as part of a team.

7. Be acquainted with the technical and human aspects of change.

8. Be acquainted with the basic principles of managing people and tasks through project management.

9. Be acquainted with how business organizations function and understand how CIS fits into this relationship.

10. Have the education and background necessary to continue with life-long learning and personal development, in addition to becoming an accomplished technician.

DERIVED REQUIREMENTS*

1. That COBOL be taught in all three required programming core courses (CIS 2,3,6).

2. That BASIC be taught in the introductory course.

3. That structured concepts be used generically in the systems courses, and that no commitment is made to any particular methodology.

4. That no more than a total of ten CIS courses be required.

5. By the 70 percent rule, no more than seven CIS courses can be in the core.

6. That four of the CIS core courses be taught at the freshman and sophomore level to minimize articulation problems.

7. That at least two of the CIS elective courses are to be taken at the junior and senior level.

8. To help newly developing CIS programs or those with very limited resources, that core courses be designated by priority of offering.

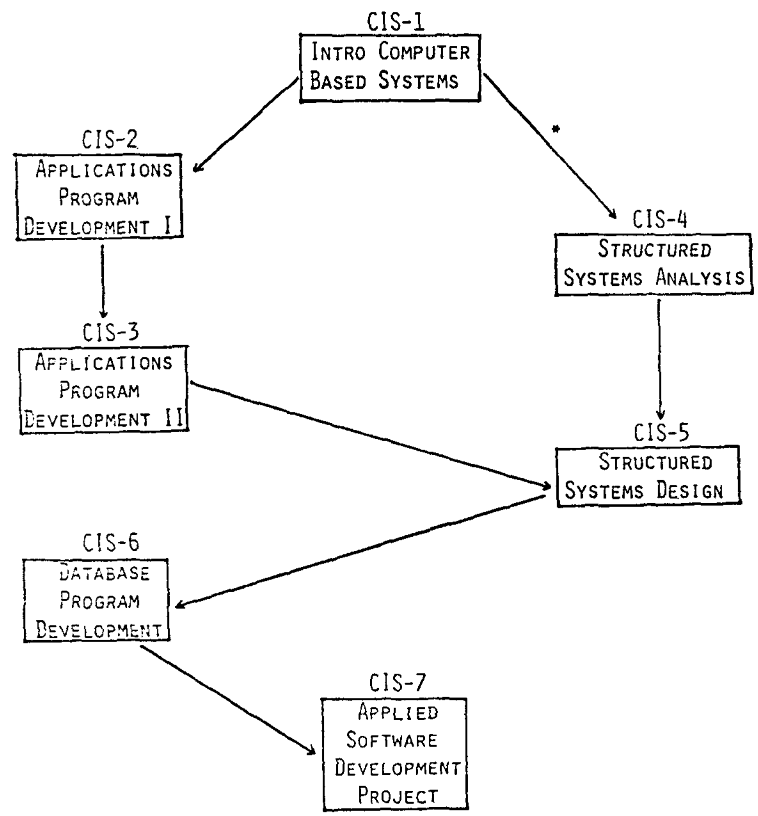
9. For the publishing community to develop the needed educational materials, course outlines must be developed as opposed to a generalized list of topics covered in the overall model curriculum.

10. That students be required to apply the concepts learned in a capstone applied systems development project.

11. The model curriculum that is developed is to be a "living" document in which future change is encourged as new knowledge is gained, the environment changes, more experience is gained with course offerings, etc.

*Derived from the 1980 Questionnaire and National Workshops.

PROGRAMMER ANALYST
MODEL CIS CURRICULUM CORE



PLUS THREE CIS ELECTIVE COURSES (SEE PAGE CE-0)

* ARROWS INDICATE REQUIRED PREREQUISITES

C-0

INTRODUCTION TO COMPUTER-BASED SYSTEMS (CIS-1)
(Freshman Level Course)

(Skill Level 1--means that the student is just being exposed to the subject matter)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Introduction to Computers	1	7
Characteristics of Computer Systems		
Hardware, Programs, Data, Procedures, People		
History of Computer Evolution		
2. *Processing Concepts*	1	15
Processing Cycle		
Artihmetic Operations		
Logical Operations		
Data Representation		
3. Input/Output	1	13
Input Media, Devices, and Applications		
Output Media, Devices, and Applications		
4. Memory Storage	1	13
Primary Types		
Secondary Types		
Advantages and Disadvantages		
File Organization		
File Updating		
5. Distributed Data Processing	1	7
Data Communications		
Centralized vs. Decentralized		
Data Bases		
6. Computer Problem-Solving Methodology	1	10
Systems Development Cycle		
Problem Definition		
Systems Analysis and Design		
Top-down Program Development		
Structured Coding		
Testing		
Documentation		
7. Programming Projects	1	25
Programming Language		
Basic Input/Output Operations		
Accumulating Final Totals		
Decisions and Looping		
Table Searching		
Simple MIS Inquiry and Reporting		
8. Future of Computers in Society	1	10
Technological Trends		
Effects of Computers on Organizations		
Data Banks and Privacy		
Computer Crime		
Employment Opportunities and Career Tracks		

Course Intent

Course is to be taught as a General Education class, not as a course for CIS majors only. The purpose is to give students an overview of the CIS field, by explaining everyday computer applications in business, medicine, education, law enforcement, etc. Students will be required to develop and test four simple programs using BASIC. Programs development should be CRT-oriented using mainframe, mini or micro computer.

C-1

APPLICATION PROGRAM DEVELOPMENT I (CIS-2)
(Freshman Level Course)

(Skill Level 2--means that the student has applied the subject matter at a medium level of complexity)

(Skill Level 3--means that the student has applied the subject matter at the job-entry level of industry)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Introduction Flowchart Symbols and Pseudo Code Using the Text Editor, The Program Compilation Process	2	5
2. Algorithms and Programs Values and Expressions, Data Types and Values Arithmetic Expressions, Output Statement	3	5
3. Variables, Assignment, and Input Names, Memory Locations, Variables, and Assignment Expressions	3	5
4. Decision Conditions, One- and Two-Way Decision Multiway Decision, Nesting, Logical Operators and Expressions	2	15
5. Looping Test Before Body of Loop Test After Body of Loop Iteration	2	5
6. Tables One-Dimensional Tables, Elements of Table Processing Searching, Internal Sorting, Two-Dimensional Tables	1	10
7. Subroutines Subroutines, Access to Arguments Local and Global Variables	1	5
8. Algorithm Design and Testing	1	10
9. Records Record Structures, Declarations Group Items and Elementary Items Input, Output, and Assignment	2	10
10. File Organization and Processing Sequential Files, Auxiliary Memory Input and Output Statements for Sequential File Processing Sequential File Update	2	10
11. Communication Skills Program Documentation, Test Data Programming Project Discussion and Review	2	20

Course Intent

This course prepares the student in the fundamental principles of computer programming. Topics include the three basic constructs of structured programming and the "building block approach" combined with symbols and the data they represent to build algorithms to effect "good" programs. The exercises include the following generic types of business information system programs, i.e., data entry, verification, reporting and file updating. The exercises also introduce the student to the various design and implementation documentation required in industry. The focus of this course is on control structures and their syntax, elementary data structures, and sequential files. A subset of COBOL is used as the programming language.

APPLICATION PROGRAM DEVELOPMENT II (CIS-3)
(Sophomore Level Courses)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Program Development Techniques & Communication Skills Top-down Bottom-up Walk-thrus Teams Testing	2	10
2. Structured Program Design Concepts Coupling Cohesion Scope of Effect/Scope of Control Effect of Data Structure and Structure Clash Transition from System Design to Program Design	2	10
3. Sequential File Processing Data Validating Control Break Logic Reporting Sort-Merge Sequential File Updating	3	.15
4. Table Handling and Variable Length Records Static Tables (Compile Time) Dynamic Tables (Execution Time) Searching/Sorting Tables	3	10
5. String Manipulation	2	5
6. Modular Program Development Subroutines Subprograms Source Program Library	2	10
7. Random Access File Organizations Relative Indexed	2	15
8. Programming Projects Data Validation File Update - Sequential and Random Control Break Logic with: Sort, Table Handling, Report Writer On-line Inquiry and Update with: Relative and Indexed File Access and String Handling	3	25

Course Intent

This course is the logical follow-on to Applications Program Development I. It prepares the student for an environment of systems of programs and the techniques for their design and development. While Applications Program Development I considered the primary packaging unit the individual subroutine, this course takes that "building block" concept up one more level to that of the program. The environment for exercise here includes dynamic data structures, random files and simple data structure modeling and implementation. Whereas, Applications Program Development I was concerned with effecting good programs, Applications Program Development II is concerned with effecting "good" systems of "good" programs. Programming projects will be accomplished by students using a full COBOL set of instructions in an on-line program development mode.

STRUCTURED SYSTEMS ANALYSIS (CIS-4)
(Sophomore Level Course)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Systems Development Process	1	10
Problem-Solving Cycle		
Top-Down Approach		
Fundamental Systems Concepts		
Problem Identification		
Gathering Information Techniques		
2. Structured Analysis Concepts	1	10
Functional Decomposition		
Process Specification		
Systems Modeling		
3. Document Current Physical System	2	20
Report Analysis		
Sampling Methods		
Systems Flow Charts		
Establish Inputs/Outputs		
4. Derive Current Logical System	2	20
Transformation from Physical to Logical		
Data Flow Diagrams		
Decision Tables		
Structured English		
Data Dictionary		
5. Data Design for New System	2	20
Conduct User Inverviews		
Develop Screen/Report Displays		
Derive Required Data Elements		
Group Elements into Logical Files		
Redesign Forms		
Human Engineering		
6. Communication Skills	1	15
Write Letters/Memos		
Interviews with Users and Clerks		
Prepare Users Documentation		
Write Systems Analysis Project		
Critique Student Projects		

Course Intent

This course is an indepth initiation to the system development life cycle, emphasizing the earlier portions. Documentation and communication aids are introduced as well as interpersonal approaches and techniques used in analysis. Products of this course include paper models of the current physical and logical systems. Accounting Information Systems case studies are used such as payroll, accounts receivables, order entry, billing, etc.

STRUCTURED SYSTEM DESIGN (CIS-5)
(Junior Level Course)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Review of Fundamental Structured Systems Concepts	2	5
Nature and Problems of Structured Analysis		
Modern System Life Cycle		
Tools of Structured Analysis: Data Flow Diagram, Data Dictionary, Structured English		
2. Analysis of Current System	2	5
Review of Feasibility Document, Review of User Requirement		
Development of Current Physical Data Flow Diagram		
Derivation of Current Logical System		
3. Planning Horizon Considerations	2	5
Environmental Impact, Organizational Concerns		
Human Behavior		
4. Model New Logical System	2	20
Translate Specifications into Working Documents		
Data Dictionary, Transform Descriptions		
Data Structure Chart, Data Flow Diagram		
Structured English, Files and Data Bases		
5. Logical Database Development	2	20
Requirements Analysis		
Logical Object Analysis		
Relational Analysis		
Aggregate Design		
6. Learn Good System Design Principles	2	5
Functional vs. Process Decomposition		
Hierarchy vs. Modular Design, Influence of Data Structures		
7. Evaluate Designs	2	5
Technical, Economic		
Psychological, Political Criteria		
8. Planned System Change	2	5
Human and Technical Aspects		
Evolutionary vs. Revolutionary Change		
Goal Congruence vs. Divergence		
Implementation Strategies		
9. Derive Physical System	2	10
Examine Environmental Requirements		
Machine Requirements, Languages, Data Base, Processers		
Complete Development of Packaged Design		
Create Test Plan		
10. Testing Design for Feasibility	2	5
Forecasting Performance, Screening by Constraints		
Internal Accounting Controls, Access Control		
Hardware Security		
11. Communication Skills	2	15
Write Project Progress Reports		
Write Intragroup and Intergroup Memos		
Verbally Present Evaluation Reports		

Course Intent

This course continues in the System Development life cycle where CIS-4 left off. Student will utilize structured design techniques to design a new logical and a new physical system for a given business related problem. The Feasibility Document and User Requirements will be provided. The student will begin in the Structured Analysis phase. Alternative new physical systems are designed and a multidimensional cost/benefit evaluation is performed. The primary product of this course is a paper model of the system to be developed.

DATA BASE PROGAM DEVELOPMENT (CIS-6)
(Junior Level Course)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Transition from Logical to Physical Database Development	3	15
What is a Database		
Objectives of a Database		
Major Approaches to DBMS		
Relational		
Network (Hierarchical)		
Implementing Data Structures		
Linked Lists		
Trees		
Networks and Plexes		
2. Third Normal Form and the Normalization Process Leading to a Stable Data Model	3	10
3. Physical Organization	3	15
Criteria Affecting Physical Organization		
Types of Access - Fetch or Search		
Retrieve		
Store		
Update		
Access Methods - Indexed & Non-Indexed		
Initial Object Access		
Navagational Access		
Overflow Access		
Single vs Multiple Record Access		
4. Survey of Commercial Database Systems	2	10
5. Practicum: Network Implementation of DBMS	3	50
Data Description Language		
Schema		
Sub-schema		
Data Manipulation Language		
Host Language Facility		
Interactive Programming - The Man/Machine Dialogue		

Course Intent

This course exposes the student to the transitional phases of the system life cycle passing from analysis through design and on to implementation. The interface with analysis is accomplished via data flow diagrams, data dictionary and structured English policy statements. The interface to design and implementation is accomplished through data structure diagrams. The implementation interface is accomplished via the data element, object, structures and storage schemas. The central focus is on complex data structure modeling and implementation. This practicum provides the student with the opportunity to work with these concepts in a simulated or real database environment (depending on the availability of a DBMS).

APPLIED SOFTWARE/PROJECT DEVELOPMENT (CIS-7)
(Senior Level Course)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Project Management Concepts	2	10
Planning and Cost Justification		
Setting Applications Priorities		
Scheduling Work, Distribution, Staffing, Contract Personnel		
Team Dynamics and Handling Conflict		
Milestones: GANTT, PERT, CPM, Spiral		
Change Control		
Risk Management		
Systems Audit		
2. Analyzing a Given System	3	10
Pre-Implementation Stage		
Hardware Limitations		
Language and Operating System Limitations		
Feasibility of Implementation		
Changes by Management since Design Completion		
3. Scheduling Implementation	3	8
Selection of Project Team		
Building Top-Down Flowchart		
Assigning Modules		
Building a Time Schedule		
4. Implementing Data Base	3	12
Designing Physical Data Base Using a Predefined Logical Data Base		
Fitting a Data Base Structure to the Hardware		
Building a Test Data Base		
5. Designing Physical Input and Output	3	7
Designing Menus for the Cathode Ray Tube		
Designing Reports		
6. Building Modules in a Top-Down Fashion	3	18
Type of Program Stubs		
Building Top Level Modules First		
Structured Walkthrus		
7. Testing Modules in a Top-Down Fashion	3	20
Testing with Program Stubs		
Interfaces between Modules		
8. Communication Skills	3	15
Verbal Presentation of System to Users		
Program Demonstrations		
Reporting to Project Manager		
Write Operating Manuals		
Write Programming Manuals		
Write User Documentation		

Course Intent

This course is set in a team environment and used to demonstrate application system development and project management. Some of the variables that were controlled in the previous courses are loosened here to approximate a "live" environment. The case study should have loose ends, holes, and outright omissions that each team will have to navigate in their passage to the development of a working set of programs. The Programming language used can be COBOL, BASIC, or PASCAL. Case study would be of moderate complexity like the development of an on-line Hotel Reservation system.

CIS MODEL CURRICULUM
ELECTIVE COURSES

CIS COURSES		LEVEL
CIS-8	Information Resource Management	UD*
CIS-9	Systems Planning	UD
CIS-10	Distributed Data Processing	UD
CIS-11	Data Base Management Systems	UD
CIS-12	EDP Auditing	UD
CIS-13	Data Processing Law	UD
CIS-14	Decision Support Systems	UD**
CIS-15	Assembly Language Programming	LD
CIS-16	Office of the Future	UD**
CIS-17	Operating Systems	LD**
CIS-18	RPG Programming	LD
CIS-19	Small Computer Software Development	UD**

*LD stands for Lower Division

*UD stands for Upper Division

**Need Course outlines to be developed for these courses

INFORMATION RESOURCES MANAGEMENT (CIS-8)
(Upper Division Course)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Future Information Systems Department	2	10
Merging of DP, WP, and Communications		
Dispersing of Processing Power		
Organizational Relationships		
Career Paths		
2. Stages of EDP Growth	2	10
Evolving Computer Systems Architectures		
Software and Firmware Trends		
Data Base Management Systems		
Decision Support Systems		
3. Information Systems Management	3	10
Operations, Programming		
Systems, Technical		
4. Developing Information Systems	2	10
Project Management		
Software Productivity		
Team Selection, User Reviews		
5. Personnel Selection, Training and Retention	2	10
Career Paths and Skills Required		
Ways to Keep People Updated		
Professionalism		
Handling Stress		
Motivation Factors		
6. Data Processing Legal Considerations	2	10
Contracts		
Data Processing Legislation		
Privacy		
Computer Security and Crime		
Proprietary Rights		
7. Distributed Data Processing	2	10
Telecommunications		
Network Architecture		
Distributed Data Bases		
Central, Decentral, and Distributed Systems		
8. Office of the Future	2	10
Data Entry/Data Capture		
Small Business Computer Systems		
Time Sharing/Service Bureau		
Word Processing, Shared Logic		
Workbenches		
9. Communication Skills	3	20
Role Playing-Manager and Interviewer		
Prepare a Resume and Critique Other's Resumes		
Attend and Critique a Professional Association Meeting or Conference		
Assess a Training Film, Seminar		
Formal Written, Oral Presentation of Research Report on Future Trends		

Course Intent

This seminar class is to be used to give students a good overview of emerging issues in the CIS field. There will be individual projects where student will research one of the topic areas and write a formal report. Student will give an oral briefing to class on research report. Students will become very familiar with literature in field, i.e., Computerworld, Datamation, ACM, Mini-Micro, IEEE, DPMA, SMIS, etc. Students will critique a training film, professional meeting, etc.

SYSTEMS PLANNING (CIS-9)
(Upper Division Class)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Planning Cycle Systematic, Integrated, Unfolding Approach Problem Identification Determining Organization Objectives Developing Alternative Solutions Judging and Evaluating Performance Selection of Best Solution Implementation Requirements	2	10
2. Determining Systems Problems Organizational Context Situational Analysis Problem Formulation and Identification User Concurrence	3	15
3. Hardware Availability Research Small Business Computer Options Sizing Requirements Financial Considerations User Interface	3	20
4. Make-Or-Buy Considerations Locating Proprietary Software Estimating Cost of Developing Software General Trade-Offs Judging Future Performance	3	15
5. Acquisition Considerations Hardware, Software, and Maintenance Screening Potential Alternatives Procurement Preparation (RFP) Evaluating Vendor Packages Judging Software Packages Dealing with Vendors Contractual Concerns	3	20
6. Decision Process Role of the Systems Analyst Presentation of Project Results to Management Group Dynamics Explicit and Implicit Objectives Selection Process	3	10
7. Communication Skills Review of Literature Observation Writing RFP Written and Verbal Progress Reports Written System Study	3	10
8. Case Studies - Small and Medium-size businesses Either first time user or a significant upgrade to a minicomputer		

Course Intent

Student will learn a systematic plan for determining a company's computerized needs. Students will learn how to research Data Pro/Auerbach for hardware options, software directories, and vendor brochures. Individuals will develop RFP's and critique vendor proposals. Students will learn how to present ideas to management.

DISTRIBUTED DATA PROCESSING (CIS-10)
(Upper Division Course)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Fundamental Concepts of Distributed Processing Centralized, Decentralized, and Distributed Systems Advantages/Disadvantages of Each Systems Components Application Areas of DDP	3	10
2. Data Communications Principles Basic Concepts Communication Channels Error Control Procedures	3	10
3. Hardware and Software Mini/Micro computers Terminals Host Operating Systems Access Software	3	10
4. Distributed Processing Networks Structures Communication Switching and Routines Network Protocols Communication Channels Computer Networks	3	15
5. Distributed Data Base Structures Evolution of Data Base Systems Data Base Requirements	3	10
6. Security Considerations Physical Security System Security	3	5
7. Implementation and Management Project Planning Hardware Evaluation Software Evaluation Measurement andd Control	3	10
8. Case Studies Retail, Manufacturing, etc. Systems Modeling using Simulation	3	30

Course Intent

This class is primarily a lecture course in which the student gets a solid understanding of what DDP is and what its advantages and disadvantages are. Data communication concepts are discussed but not at an operational level of detail. Applications are covered in case examples and the principles of simulating a network design are covered.

ADVANCED DATABASE CONCEPTS (CIS-11)
(Upper Division Course)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Database Technology	2	30
DBMS Environment		
Hardware - Device Characteristics and Constraints		
Software - Operating System		
Physical vs. Logical Input/Output		
Input/Output Supervision		
Data Communications		
DBMS Organization		
Data Description		
Data Manipulation		
Host Language Facility		
Overy Facility		
Utilities		
2. Database Administration	2	30
Data Dictionary and Other Tools		
Social Impart - Security, Privacy, Integrity		
Concurrent Processing		
Backup, Recovery and Transition Logging		
Performance Monitoring and Tuning		
Distributed Processing		
Redundancy		
3. Selection and Acquisition of a DBMS	2	20
4. Future Trends in DBMS	1	10
CODASYL		
Relational Models		
Standardization		
5. Indepth Comparison of Commercial DBMS	2	10

Course Intent

This course addresses the two major areas of Database administration and Database Technology. Individual projects are assigned based upon students interest in either area. All students will participate in a Selection and Acquisition Exercise that will involve indepth investigation of some commercially available DBMS.

EDP AUDITING (CIS-12)
(Upper Division Course)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Computer Based Informations Systems	3	10
Controls and Auditing		
Computer Abuse and Impact		
2. Systems of Controls	3	25
Administrative		
Operational		
Documentation		
Security		
3. Computer Audit Techniques	2	25
Basic Computer Audit Techniques		
Computer Assisted Audit Techniques		
4. Auditing Real-Time Computer Systems	2	20
Event Processing Controls		
Distributed Processing Systems		
Electronic Funds Transfer System		
5. Systems Approach to Auditing	3	20
Risk Assessment		
Threat Analysis		
Cost/Benefit Analysis		

Course Intent

This course covers the why and how for auditing computer-based information systems. The strengths and weaknesses of auditing software packages are discussed. Students should have access to a commercially available audit software like STRATA, TREAT, etc. A systems approach to making an audit assessment is covered.

DATA PROCESSING LAW (CIS-13)
(Upper Division Course)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Research Techniques Sources of Law (judicial, administrative, legislative) Sources of Information (government publication, computer-related journals)	2	10
2. Proprietary Rights Patents, Copyrights, Trade Secrets, Trademarks Statuatory Regulations Software Firmware	3	15
3. Contracts Hardware and Software Maintenance Consulting Warranties Liabilities	3	15
4. Privacy Freedom of Information Act (1966/1975) Privacy Act (1974) Fair Credit Reporting Act (1970)	3	15
5. Computer Security and Crime Risk Assessment "Ribicoff Bill" (S. 240) State Crime Laws Criminal Cases Legal vs. Ethical Standards	3	15
6. Law, Competition and Computers Antitrust Cases Pricing Regulation Unfair practices	2	10
7. Data Processing Legislation Foreign Corrupt Practices Act Data Processing Crimes Act Electronic Funds Transfer Act	3	20

Course Intent

The purpose of this coursse is to acquaint the student with most of the important legal issues in data processing. Research reports and case studies are used to highlight these concepts. Further students are introduced to the computer law literature.

ASSEMBLY LANGUAGE PROGRAMMING (CIS-15)
(Lower Division Course)

TOPICS	SKILL LEVEL	PERCENTAGE OF TIME
1. Computer Hardware Architecture Central processor, memory, registers Addressing method, program status words Branching and interrupts, data storage Techniques zoned, packed and binary data	2	25
2. Instruction Set Instruction formats, operation codes, operands Machine and assembler	2	20
3. Manufacturer Software Operating system, linker, programs cataloging Translations and compilation Memory maps and dumps	2	20
4. Programs Basic Loop Conditional and unconditional branches Sub-routine linkage Indexing tables Address modification	2	35

Course Intent

The purpose of this course is to give students more depth in understanding how computer software works and the impact of various hardware architectures. Further, the student will be shown the efficiencies that can be gained using a lower level language.