DOCUMENT RESUME

ED 082 509 EM 011 482

AUTHOR	Burns, Harris, Jr., Ed.; And Others
TITLE	The Administration and Management of Small-College Computing Centers. Proceedings of a SIGUCC Symposium (Atlanta, Georgia, June 9-10, 1972).
PUB DATE	Jun 72
NOTE	77p.
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	Business Administration; *Colleges; *Computers; *Computer Science; Computer Science Education; Conference Reports; Electronic Data Processing; Management Development; Professional Training; Programing Languages; Symposia; *University Administration; Use Studies
IDENTIFIERS	*Campus Computer Centers

ABSTRACT

The small college computer center's problems--sources and types of service, programing support and faculty development, administrative and academic interfaces, computer science and computer center interfaces, computer center policies, orientation and organization and services of users and groups--are generally individual, and any given problem will have many alternative solutions. This report of a symposium on small college computer centers presents papers and addresses which are oriented around this theme, with each set of papers being presented according to one of the problems listed above. In addition, the symposium's program, an introduction to the symposium, and its keynote address are included. (Author/SH)

THE ADMINISTRATION AND MANAGEMENT OF SMALL-COLLEGE COMPUTING CENTERS

Proceedings of a SIGUCC Symposium

June 9-10, 1972

Sheraton-Biltmore Hotel
Atlanta, Georgia

Edited by

Harris Burns, Jr.
Randolph-Macon College

Gerald L. Engel
Pennsylvania State University

John H. Esbin
The University of Iowa

THE ADMINISTRATION AND MANAGEMENT OF SMALL-COLLEGE COMPUTING CENTERS

Proceedings of a SIGUCC Symposium

June 9-10, 1972

Sheraton-Biltmore Hotel
Atlanta, Georgia

Edited by Harris Burns, Jr.
Randolph-Macon College

Gerald L. Engel
Pennsylvania State University

John H. Esbin
The University of Iowa

Forward

This volume contains the papers presented at the SIGUCC Symposium on the Administration and Management of Small-College Computing Centers held at the Sheraton Biltmore Hotel, Atlanta, Georgia, June 9-10, 1972.

Though the material presented here does not represent a blueprint for successful operation of a small college computer center, careful study of the papers will reveal many solutions to problems most directors face. A unique feature of this volume is that the problems and solutions are given by people who have been there. For the most part the material does not represent philosophic views of individuals looking down from on high.

As in any meeting there are many acknowledgments that need to be given; to the SIGUCC Officers and especially Gordon Sherman for encouraging the creation of the meeting; to R. Waldo Roth and George Heller for their service on the planning committee; to John Hamblen for handling local arrangements; and, of course to the speakers and attendees without whose assistance the meeting could never have been held.

Gerald L. Engel
Pennsylvania State University
General Chairman

Harris Burns, Jr.
Randolph-Macon College
Program Chairman

TABLE OF CONTENTS

THE COMPUTING CENTER AT THE SMALL-COLLEGE: THE SENSE OF THE MEETING

Gerald L. Engel
Pennsylvania State University

The forces have met; they have agreed they have problems; they will meet again.

Such is the sense of most meetings and indeed this one was no exception. What was an exception was that some fifty individuals with the common experience of running small computer centers on small budgets got together to compare notes. This was a meeting of those who had been through it, and those that are in it, not a meeting of high beliefs of how things should be.

In my opening remarks as Symposium Chairman, I characterized the small college computer center director as part teacher, part registrar, part dean, part business manager, part janitor, and part football coach. Nothing I heard in Atlanta changed my mind on this, and instead my respect for these individuals, who will give so much time of themselves to bring adequate computing services to their campuses, is justified.

What did we discuss? The topics were

Sources and Types of Service
Programming Support and Faculty Development
Administrative/Academic Interface
Computer Science/Computer Center Interface
Computer Center Policy
Orientation and Organization and Services of Users and Groups.

What did we learn? That given any problem there are many possible answers.

But to characterize the meetings by the sessions and the speakers is a mistake. The heart of the meeting was the contact with people with the same problems. It was the swapping of programs and materials that went on after the formal sessions had concluded. It was the knowledge of sources of aid that could be called on when problems come up in the future.

All types of schools with all types of facilities were represented. The common bond was that they had not had a forum for airing their views.

If there was a common ground, in a sense, to the meeting it was this. The small college computer center is a very individual beast. There are no magic formulae that will solve all the problems associated with its operation. Instead as many alternatives as possible must be considered and from these the best choice must be selected.

From my experiences at the meeting, I am convinced that the small schools have the people that make the best choices of alternatives, and it is my hope that this meeting, and future ones like it, will prove to be the place where the alternatives are presented and discussed.

PROGRAM

Friday, June 9

10:00 - 10:30 Opening Remarks*
Gerald L. Engel, General Chairman
Harris Burns, Jr., Program Chairman
Gordon Sherman, SIGUCC Chairman

10:30 - 11:15 Introduction
Preston Hammer, Pennsylvania State University

11:15 - 12:00 Keynote Address
W. Taylor Reveley, Hampden-Sydney College

1:30 - 3:00 Sources and Types of Service
Chairman: Richard Austing, University of Maryland
Aaron Konstam, The Lindenwood Colleges
Fred Weingarten, The Claremont Colleges
Bruce Alcorn, National Laboratory for Higher Education
Whitney Johnson, Virginia Council of Higher Education
Discussant: Glenn Ingram, Washington State University

3:30 - 5:00 Programming Support and Faculty Development
Chairman: Harris Burns, Jr., Randolph-Macon College
Louis Parker, North Carolina Educational Computing Service
Pamela McGinley, Technica Education Corporation
Karl Zinn, University of Michigan
Discussant: Fred Weingarten, The Claremont Colleges*

Saturday, June 10

9:00 - 10:30 Administrative/Academic Interface
Chairman: R. Waldo Roth, Taylor University
Jacques La France, Wheaton College
Ronald Anton, Swarthmore College
Discussant: Bruce Alcorn, National Laboratory for Higher Education*

11:00 - 12:30 Computer Science/Computer Center Interface
Chairman: Harris Burns, Jr., Randolph-Macon College
Jesse Mayes, Federal City College
Robert Kyle, Emory University
Richard Austing, University of Maryland
Discussant: R. Waldo Roth, Taylor University*

1:00 - 3:30 Computer Center Policy
Chairman: Bruce Alcorn, National Laboratory for Higher Education
Richard Vogel, Western Maryland College
James McDonald, Morningside College
Discussant: Harris Burns, Jr., Randolph-Macon College*

3:30 - 5:00 Orientation on the Function, Organization, and Services of Users' Organizations*
Chairman: R. Waldo Roth, Taylor University

*Remarks not included in the Proceedings

INTRODUCTION TO THE SYMPOSIUM

COMPUTER SCIENCE AND SMALLER COLLEGES

Preston C. Hammer
Chairman, Computer Science Department
Pennsylvania State University

Of the many problems facing smaller colleges, the most difficult may well be the selection of a philosophy adequate for these changing times. While financial difficulties cannot be disregarded, these are probably easier to understand than it is to provide educational opportunities which attract students. I have in the course of the past twelve years visited over 100 colleges. In many of these I was asked what the college should do about the hyperactive computing field which was customarily regarded as a threat by the mathematics faculty. The major handicap seemed to be in the interpretation of the role which computer science education might conceivably have in a liberal arts college. The interpretations of computer science amounted to regarding it as vocational training rather than as a possibly vital component of a liberal arts education. Being a graduate of a liberal arts college myself, I have some idea of the ideals which such colleges have. I shall accordingly address myself to the roles which computers and computer science might play in colleges and, conversely, what the colleges may contribute to this new area.

COMPUTER SCIENCE AND RATIONALITY.

Modern electronic computers are time amplifiers. They are, in common with all our tools, extenders of our capabilities. Where they differ from such scientific apparatus as microscopes, telescopes (space amplifiers), accelerators, and other gear is in the universality of their applicability. In virtually every area of organized human activity there are at least some few individuals making use of computers. Applications are being made in health services, natural sciences, engineering, social sciences, architecture, fine arts, music, historical studies, businesses, governmental agencies, colleges and universities. In some of these applications computers do tasks which could be done manually but, in the best applications, tasks are being done which were not possible before.

The versatility of electronic computers is, in large measure, due to their incompleteness. That is, the computer enables a large number of specific tasks to be done by following instructions provided for it by people. Thus a computer is not simply one machine; it is a possibility of myriads of machines depending on the ingenuity of its human users for the effectiveness of the help it provides.

The computer is best at repetitive tasks. It can carry out millions of arithmetic operations and make millions of decisions which it has been instructed to do. It will also carry out manipulations with symbols with great speed when properly instructed. Accordingly the most used branch of mathematics, arithmetic, achieves vastly increased usefulness when computers do the work. Computers, when provided with proper software, can carry out formal differentiation better than humans. However, people have to do the planning and provide the software, even as other people had earlier designed, made, and distributed the hardware.

The utilization of computers then requires study of strategies to reduce classes of problems to computability. For example, I have stated and proved a rather large number of theorems. If computers are to be used effectively in proving theorems, then a study must be made of the strategies and flows of proofs in each area where they are to be used. Now, on the working level, I might claim to know how to both generate and prove theorems. Yet I do not know a strategy of proof which I can convey to a computer. While working on strategies of proof is a higher level of activity than proving theorems, this does not mean that individuals who try to mechanize proofs are somehow superior to those who ably prove theorems. However, I anticipate that there will be some of the best mathematical minds of the future developed in the attempt.

In some ways, the computer provides a means of separating the rational (i.e. computable) from the irrational. Any process which we comprehend thoroughly might be simulated in a computer, but those which we do not comprehend resist reduction to a computer. The problem of mechanical translation of languages was once deemed a matter of doing a rather moderate amount of work to achieve success. The attempts proved that language is not that simple, and a new appreciation of linguistic structure has been developed as a byproduct. It was once thought that computers would make great strides in meteorology possible. Today we must admit that the weather is too poorly understood and too complex to be well predicted by any yet known means.

The computers are poorly suited for recognizing geometrical patterns. Biological problems are generally beyond their scope. It is easy to state medical problems for which solutions would be desirable which are simply too expensive to solve now. It is easy to give simple mathematical problems which would be beyond the reach of computers even though, in principle, they would so be managed.

Thus the presence of a computer, which may be and has been used as a means of escaping the rigors of thinking, actually encourages greater use of the intellect by the promise of amplifying its product.

There is scarcely a more rigid discipline than programming for computers. Very slight errors lead to completely erroneous results as the machines follow instructions exactly. One mathematical logician has pointed out that the discipline of programming is excellent preparation for study of logic. Conversely, a grasp of formal logic is a useful background for effective computer programming or computer design.

To put this in the context of this paper, there is much in favor of having computer science in undergraduate programs in liberal arts colleges.

COMPUTER LANGUAGES.

Having been responsible for the initiation of two computer science departments, and having a background in mathematics, I have mused on the necessity of establishing separate departments. The immediate necessity arose because of the necessity of teaching computer languages and software design. In general, the electrical engineering departments do not teach non-engineering students, and the teaching of computer languages was placed on the same plane as teaching English. Mathematicians in universities, on the other hand, while accustomed to teaching service courses, generally would not take computer languages seriously.

Accordingly it was necessary to start a new discipline. Almost everyone admits that the learning of foreign languages is best started when young. The same is true of computer languages. Whether the learning is for the objective of a career in the computer profession or for applications in some area, developing real facility with computers requires more than casual efforts.

Despite several sporadic efforts the computers have not yet been effectively used in undergraduate education in courses in most areas. This is not because it would be ideally impossible to make use of computers, but because of the major revisions of courses which would be needed. Wherever the computer is used effectively, a drastic revision of educational procedures is necessary.

I think that liberal arts colleges could take the initiative in experimenting with educational uses of computers. This belief is founded on the dedication of faculty to education as compared to the research emphasis of universities.

Moreover, the cultural aspects of computers are certainly more likely to be clarified and presented by scholarly educators rather than by technical research workers.

PRACTICAL ASPECTS OF COMPUTER SCIENCE.

While some colleges may not admit it, all colleges of which I know are basically career preparation oriented. Rare is the student these days who feels he or she can afford to ignore planning beyond the baccalaureate degree. Communication skills are acquired and communicating with computers is becoming increasingly useful. Among the sciences and mathematics there is no discipline which prepares a student for employment as well as computer science does at the baccalaureate level. For almost all areas of graduate study and research, prior study of computers is an asset, with the possible exception of pure mathematics.

Why is computer science so practical? Because, what a student learns in one or more courses is almost all used in any computer applications the student may later make. In contrast, how much of other courses really is that effective?

THEORETICAL ASPECTS OF COMPUTER SCIENCE.

The theories needed for the study of computer science are mathematical in aspect but the emphases are different. Constructive logic in general, recursive functions, and algebra structures of logic achieve a relevance in computer science which was not noted before. Arithmetic and algebra are essential. Automata theory and formal linguistics are defined axiomatically. Graph theory and combinatory analysis are of concern in analyzing algorithms and organizing programs.

How about such mathematical classics as geometry and calculus? Despite the tendency of many computer scientists to belittle these areas in favor of discrete mathematics, both are important in computer science. The calculus and analysis provide continuous models of real systems and continuous implication systems (logics) to deduce their implications. Since only a few of the practical problems of analysis are solvable effectively in closed form, numerical analysts replace continuous logics with discrete logics and enable more solutions via computers. Each real system as we idealize it is embedded in a four-dimensional space, but usually the number of its parameters places it in a higher dimensional space. The forms and shapes of systems are usually represented geometrically. Hence geometry should not be ignored.

However, does computer science actually need any but extant theories and interpretation of mathematics? The answer is that much important concepts, as that of function, take on new life and meaning in computer science. For example, I pointed out three years ago that the customary synonym "transformation", used for function, is better replaced by "transformer". This idea came to me when I was interpreting a computer as a function which transforms input (program) into output. A computer is <u>not</u> a transformation of inputs; it is a <u>transformer</u> of inputs.

This means that from elementary algebra on, identities have been misinterpreted in mathematics, since they confuse the result with the process which produces the result. In computer science two algorithms which produce the same results are not to be regarded as equivalent completely! In other words, the distinction between a verb and the object in a sentence is relevant in computer science.

It was in using so-called random number sequences in computation that I jelled my conclusion that randomness is nonsense, which it in fact is. Moreover, I have recently written an essay on the relationships between algebras and geometries. These relationships are necessary to computer science but have never been clarified in mathematics. Why are they necessary in computer science? Because each computer has only a finite number of symbols available, and it is necessary to use these symbol structures to represent systems. But the language (algebra) of geometry is finite, and so is the language of analysis. It is more important in computer science, then, to find out how these algebras work, than it has been in mathematics.

Now in my estimate, the situation of computer science with respect to mathematics is that computer science is restoring the dynamic and culture-sensitive aspect which was disappearing from mathematics. It is much easier to use the cook-book formulas of calculus than it is to design effective programs for computers. On the other hand, a thorough knowledge of mathematics is needed for many aspects of computer science. Small abstract systems may be manipulated by computers in ways not possible by man. Patterns in such systems are still hard to discern using computers. Finally, computer science provides one of the best filters for current mathematical effectiveness; it shows the limits of our knowledge with disconcerting sharpness.

CONCLUSION.

Liberal arts colleges can do much to save computer science from becoming a merely technical specialized field. This capability rests, in part, on the usual dedication of scholar educators, and in part on the mental energies of the students. Since the computers will enable new symbol structures, such structures may well be considered in liberal arts colleges. The research oriented college teacher may find that analysis of data, collection of data, and control of experiments can be helped by computers. However, such teachers should not be swayed toward trivial work, seduced by computers, but use computers to produce better results when it can be done.

KEYNOTE ADDRESS

THE COMPUTER CENTER IN THE SMALL COLLEGE

W. Taylor Reveley, President
Hampden-Sydney College

William N. McBain engaged in a bit of crystal gazing as he contemplated the future of higher education in an article entitled "Education Ex Machina".[1] His hero, Jack, marched to college; exchanged a number of dollars for an equal number of tokens; selected the course he would study that day; and was assigned to an appropriate booth. There, after introducing himself to the computer, he proceeded from where last he had been in the course.

The computer analyzed his performance and set the pace for his further study. Jack placed a token in the proper slot, and the computer proceeded to the next question. If he answered that question correctly his token was returned, and he deposited it again for the next question. When he did quite well on a series of questions, Jack received more tokens than he had deposited. Further, in addition to the "jack-pot" incentive, he periodically received a special token permitting him to have an hour's conference with the professor. The particularly able "Jack" could earn his way through college and also spend many hours in close discussion with his professor.

The author elaborates the educational advantages of such instruction:

> "The writing of a program forces upon an instructor the necessity of logical development. The analysis of errors makes his lapses self-evident, while the common discipline of this approach to instruction makes comparable standards more likely. The classroom boredom as well as the frequent absenteeism of the brighter student is reduced, as is the lagging and consequent bewilderment of the duller one. Indeed, there is some reason to believe that the laggard student is not more stupid, but only less quick. Failing the development of the long touted 'smart pill', it may be that all students permitted to achieve at their own rate (a possibility never before realized in actuality) will eventually arrive at a higher plateau than previously believed possible."[2]

This Orwellian glimpse into the future probably does more to frighten contemporary academicians with the spectre of blinking, whirling, giant-sized computers gradually engulfing the campus than it does to document the educational validity of computer techniques. It is somewhat like the TV show which described the university computer in the hands of an unscrupulous but brilliant technician. Four students had manufactured an imaginary person complete with all necessary credentials in order that they might use "his" credit card. The technician, in order to appropriate the identity of the person thus created, used the computer to murder three of them before his evil machinations were discovered and he was unmasked.

I cite these two approaches to computers to illustrate academic and popular amazement, wonder, and, yes, even fear of the computer. Its voracious capacity to absorb data, rearrange the relationships of those bits of information, and regurgitate with split second timing vast quantities of hitherto inaccessible calculations and collations, is enough to frighten the stoutest heart. The high priests who tend to its needs, constantly feeding its insatiable appetite for data and closely guarding its hoard of mystic knowledge, increase the mystery for the uninitiated and extend the fears so that the very mention of the name "computer" strikes fear in the hearts of all who hear. Thus, when Hampden-Sydney first secured a computer and that fact was proudly announced at the opening convocation, one student rushed to me quite perturbed. He did not want to have the college own a computer, for he felt that the long-cherished personal qualities of Hampden-Syndey's educational pattern must inevitably be extinguished.

Perhaps the most important aspect of the irrational fears aroused by the computer on campus is the witness of those fears to the tremendous importance of this new instrument of modern technology. To use the term "new" in describing the computer may seem strange in the academic scene of 1972. Statistical analysis suggests that at least 2/3 of the colleges and universities in the nation have some form of computing facility. A manual prepared by the Association of American Colleges entitled "Does Every Campus Need a Computer?" seeks to inform the small college of the virtues and vices of adding a computer to their program. It points to the fact that the "Use of computers on the campus has passed the innovative and status phase. The larger universities are so committed to their use that they could not function without the assistance of computer programs. Many of the larger liberal arts colleges have now arrived at the same position."[3]

Yet less than a decade ago a book, which was published in 1967 in fact, points proudly to the fact that "more than 1000 are installed on approximately 500 campuses".[4] However, at that period curriculum offerings, although proliferating rapidly, demonstrated "little uniformity of opinion as to what represents adequate academic training in computer or information service".[5] Furthermore, "only a very few universities in the country ....have faculty members prepared to teach in each of these areas, and are most fortunate if they can cover two or three."[6] Under the auspices of the ... tremendous progress has been made in bringing curricula order out of the chaotic diversity of the earlier years of the 1960's. Even so, the path of the small college in exploiting the academic features of the computer continues to be obstructed by economic, organizational, and intellectual barriers.

Let me turn to a more positive statement concerning the importance of the academic uses of the computer. One writer compared the cost of the computer operation to that of the library. He found that computer budgets in the institutions under survey amounted roughly to about one half the amount spent on library budgets.[7] Another justified such an expenditure by pointing out that:

> "It is safe to predict that within the next ten years a major college or university which does not have adequate computer facilities will be considered as unattractive as a college without a library... One even begins to hear it said that in humanities and liberal arts programs there is a growing realization that the citizen of tomorrow had better know as much as he can about computers."[8]

That last comment strikes home. For it suggests that computer science must appropriately find a place within the spectrum of the liberal arts. I sense the importance of this occurring not only for the integrity of the liberal arts, but also for the integrity of computer science.

If the role of the liberal arts curriculum can be found in preserving the great ideas of mankind, in understanding man, in preserving his humaneness, then it must follow that a discipline which speaks a language open to all men, which forces men to respond with precision and clarity, which touches almost every other discipline, must find a place in the liberal arts. It does not seem too extravagant a demand to insist that some awareness of the nature and function of computer science be a part of the experience of any liberally educated man. Just to know that one is to return the strange looking card without punching, bending, or mutilating is not enough.

By virtue of its impact on all other disciplines, computer science remains essentially a service discipline, and one of amazing proportions. Professor Richards describes this role well:

> "The outcomes of technological revolutions are probably unpredictable. Foresight, at least, was in short supply as to what would happen with each of the sudden unheralded accessions of power which have divided the last three centuries into periods more charged with hope and despair than even those marked by the coming of agriculture, metallurgy, or writing. What was done with steam? With electricity? With the automobile? With radio and television? What has been done with the airplane? What is being done with nuclear resources and missile capacities? What is likely to be done with weather control? It is evident that immense transformations of human possibility that may look like blessings can come to seem more like afflictions. Why should we think that a means to the increase of human power, in many ways surpassing and transcending all of these together, will in fact be more intelligently, more humanely, and more wisely used?
>
> All the foregoing epochal steps may be regarded as extensions of familiar specific capabilities; steam replaced and transcended men's and horses' muscular energy as photography and telephony surpassed and extended the range of our distance receptors. So, more widely, did radio and television. But the offerings of the computer go beyond all such services; they extend the resources of the central nervous system itself. The computer can supply an inexhaustible slave service for whatever we have the wits to instruct it to do. Suddenly, we have a Caliban-Ariel executive that will achieve for us all that we, in our wisdom or folly, can contrive to tell it how to handle.
>
> Someone will reply that computers, by taking immense intellectual burdens off our shoulders, will free us for precisely these tasks of ultimate choice, these legislative acts. We may hope so, while fearing that they will not. Almost all of us are products of the assistance we can accept. Equally, we are potential victims of those who, for whatever motives, would like to run things for us. Like all power sources, the computer is not going to lessen our responsibilities but to increase them."[9]

Because the computer performs primarily as a servant to knowledge rather than a creator to knowledge, the chief academic function of the computer center must consist in preparing the way for other disciplines to achieve new and hitherto impossible advances. The role of servant in this sense represents no mean calling. Indeed, the computer programmer must needs be, or soon become, a liberally educated man if he is to fulfill the promise of his position in melding the language of the computer with the language of each of a wide variety of disciplines.

Professor Roberts selects another but not too dissimilar analogy to describe this relationship between the computer expert and those with whom he is associated. The programmer is like Plato's guardian who has come out of the cave, where men are subject to the illusion that shadows playing on the cave wall are true reality, and has reached the bright sunlight of reason wherein the nature of true reality can be observed. Now he, the guardian, must enter the cave and seek to instruct those who mistake shadows for reality.[10]

This is heady business -- enlightening the enlightened! The computer programmer must not fall into an overweaning confidence in the ultimate absoluteness of his position any more than he can permit the tenacious proclivity of most professors to resist any change which forces them to reevaluate the entire base of their approach to the data of their discipline. He remains a servant. But it is not a service to the illusions of even a learned person's opinion. Rather his is a service to the quest for truth. To that quest, in whatever discipline involved, he brings a tool possessing vast capabilities to extend the reach of human thought. The computer programmer more often than not will meet initial resistance to his efforts to serve, and must be sufficiently knowledgeable and persuasive to convince the most timid of the seekers after truth.

To have argued for the imperative of a service role in computer science in the liberal arts college by no means is to deny the validity of computer science as a discipline in itself. In point of fact, that validity has never been in question. The problem instead has been the danger of becoming so enamored with the discipline that the service role has been ignored. This is why computer science needs the liberal arts college. This is why such colleges must look with careful concern before instituting a department of computer science. It becomes enticingly easy for the members of the computer science operation to be absorbed with just the exciting possibilities of the field itself and with those students who share their interests. Important as this absorption may be in the large university for the progress of the discipline, it should not, even there, encompass the entire attention of the computer center. In the small liberal arts college, where demands of economy restrict the number of computer personnel, the service aspect must take precedence.

Organizationally care must be taken to assure that the overall direction of the computer program remains in the hands of a person with a commitment to the liberal arts. In general, although not always in particular, the achievement of this organizational goal means that the director of the center should not be at the same time the primary professor of computer science.

Probably, withholding the establishment of departments of computer science can not be long continued; but when necessity or wisdom force the establishment of the department, it should be because the service aspect of the computer has become so strongly established that other departments are demanding greater sophistication in the training given on the campus. The demands of graduate schools for better prepared students will also push the college in the same direction.

Earlier reference has been made to the amount of the educational and general budget which can appropriately be assigned to the computer operation; namely, one half of the amount spent on the operation of the library. In the manual "Does Every Campus Need a Computer?" (p. 16) a survey is reported which indicated that in some instances cost ran as high as 5% of the educational and general expenditures. Although it was quickly added that "colleges now above the 5% level should be sure they are getting enough from their computer activity to justify the apparently excessive cost." When one recognizes that the approved percentage for library costs in the Southern Association is only 5%, the wisdom of the warning seems quite realistic.

However, when one speaks of 2 1/2% or 5% of the educational and general budget, the moment of truth has come. It just does not make sense to embark on a program involving a computer until the college stands ready to make a substantial commitment of its resources to the undertaking. And having once begun the operation, there can be no turning back the costs without fatally harming the program. While no statistics have come to my attention, and possibly because no statistics have been widely circulated, it seems likely that very few colleges have instituted computer service and then withdrawn from the program -- another witness to the effectiveness of the program on the educational process.

The problem arises in a different way. Seldom will the question be "shall we close the computer center?". Annually it will take the form of "Why can we not make our program more sophisticated?". To do so, a memorandum from the computer center will succinctly explain, requires only the substitution of a two-disk drive for the present single disk currently used. This will cut operational time in half (from 6 seconds to 3 seconds) and it will also make it possible to lease a different printer operating at quadruple the speed of the printer now in operation. The cost will be only $700 -- a month, that is.

Parenthetically, I have always had a harsh feeling for the manner leasing prices are quoted. $700 doesn't sound so terrible until stuck beside "per month". Of course, no one will or can rent the machine for only a month, or even 9 months. It has to be a year. There seems to be far less intent to deceive in the straightforward price of $8,400 annually. For the average small college computer budget ($40,000) that means a 20% increase in cost! And that when faculty salaries struggle to make 5% and everything else is being reduced or held steady at best!

The terrible part about the memorandum from the computer center is that it is undoubtedly correct. The level of usage, qualitatively at least, has reached that sophistication which makes the present operation woefully inadequate. New users and creative programs are crying for equipment sufficient for their mass and complexity.

In most cases the answer to the memorandum cogently outlining the computer center needs will be negative. This denial will bear no malice; nor will it reflect a frightened and suspicious attitude on the part of the dean, the business manager or the president. It will simply reflect the fact that there are no priorities for increased expenditures in most small college budgets; and even where possibilities for increases do exist, they will seldom arrive in 20% patterns.

I speak of such dismal realities not to discourage but to encourage you. The president may indeed be an irascible, short-sighted obstructionist; certainly these are recognizable adjectives in every president's experience. No doubt they frequently ring with truth. When, however, the computer center comes for decision, there are few presidents (the colleges of the few are probably deep in debt and running a deficit) who will not feel a strong commitment to support and strengthen the operation of the computer center.

In those colleges where funds could be made available and where the administration has a strong commitment to the computer program, the battle is only half won. The second part of victory must be won in the computer center. There the functional operation of the center must embrace the purposes of the college and the purposes of the computer department if they differ. This embrace is no private affair but must be publicly made and draw all of the college academic constituency to share in the embrace, at least all who will. In short, the computer program can be no esoteric exercise for the initiated few who will spend long hours pursuing personal fancies and racking up "computer usage" points for later reports.

The last part of the victory to be won in the center itself is the quality of the memorandum. Most college deans and presidents have sought to acquaint themselves with the functions of the computer -- and most of them have been unsuccessful! Still, even in such an uninformed state it is not difficult to distinguish a carefully planned memorandum from an inadequately conceived one. The need must be documented -- it is not enough to say a new gadget has come out and the computer staff wishes to experiment with it! The cost must be analyzed to demonstrate that the proposal made represents the most efficient use of the college's money. The persuasive line of one able salesman may have sold the computer director, but it is not enough to convince any tough minded business manager. When all of this has been done and the money is available, chances of success are high. However, should a negative response still be given, one more gambit remains to be played.

Correctly administered, the computer center becomes a strong power center on campus. The computer director can issue orders to even the most revered of professors. If administrative work falls within his purview, he can boss everyone on the staff. Naturally, then, the administrators harbor a sneaking suspicion that the power displays in the computer center represent more than judicious and objective utilization of power. "Empire building" justly represents a dangerous threat, especially on a small college campus, since empires on a campus are no less bellicose than those which keep the world in almost constant strife. If the annual expansion memorandum has been turned down, the smart computer center operator beats the administration to the next tactical maneuver. He suggests the need for an objective consultant to be brought in from outside. His presence and his report can remove the suspicion of a power play and can also give the administration something to wave in front of the finance committee of the board.

I have not spoken of and will not speak at any length about administrative use of computers. Acknowledgedly, such use does not save money either. It can give the extra precision and information in many areas. In some others, manual operation continues to remain the most effective.

[1] McBain, William N., "Education Ex Machina", <u>Liberal Education</u>, Vol. LV, No. 4, December 1969. pp. 497-504.

[2] <u>Ibid</u>, p. 503.

[3] Association of American Colleges, <u>Does Every College Need a Computer?</u>, 1971, p. 2

[4] Bushnell, Don D. and Allen, Dwight W., Editors, <u>The Computer in American Education</u>, New York: John Wiley & Sons, 1965. p. 156.

[5] <u>Ibid</u>, p. 157

[6] <u>Ibid</u>, p. 159

[7] <u>Ibid</u>, p. 157

[8] <u>Ibid</u>, p. 220

[9] <u>Ibid</u>, p. xvii-xviii

[10] <u>Ibid</u>, p. xxvii

SECTION I: SOURCES AND TYPES OF SERVICE

Chairman: Richard Austing, University of Maryland

THE SMALL COMPUTER: PRO AND CON

Aaron Konstam, The Lindenwood Colleges

INSTITUTE FOR EDUCATIONAL COMPUTING

Fred Weingarten, The Claremont Colleges

COMPUTERS IN SMALL COLLEGES

Bruce Alcorn, National Laboratory for Higher Education

COMMENTS ON SOURCES AND TYPES OF SERVICE

Whitney Johnson, Virginia Council of Higher Education

DISCUSSANT'S COMMENTS

Glenn Ingram, Washington State University

THE SMALL COMPUTER: PRO AND CON

Aaron Konstam
The Lindenwood Colleges
St. Charles, Missouri

1. INTRODUCTION

A university computing center is unique in the diversity of users it supports. It must serve the research and instructional needs of the faculty as well as the information processing needs of the college administrators. Each type of user makes different demands on a computer installation, and for each type of user a different computer system would be best suited. To a faculty member, easy access to the computer by means of terminals and time sharing would be far more desirable than faster memory cycle time or speed of input-output. Administrators would shutter at being required to transfer information in and out of a computer at terminal speeds. Few commercial computer service bureaus are faced with users less compatible from a computer service point of view. These incompatibilities, as well as the companion problem of service priorities, are continually plaguing the university computer center.

The computer center director faced with this diversity of needs is continually performing his juggling act of services so as to offend the fewest people, and periodically he is confronted with the problem of choosing equipment to optimize his services. In this paper we discuss the advantages and disadvantages of the use of a small computer to handle the diverse needs present in the college environment. In particular, we will treat the question of the use of a small computer to meet the computing needs of the small liberal arts college.

2. THE SMALL COMPUTER

At this point the question arises of what is meant by a small computer. At the Lindenwood Colleges the small computer is an IBM 1130 Model 2B, with 8K of memory, one disk drive, a 1132 line printer and a 1442 card reader/punch. More generally, we will be speaking of computers that operate mainly as dedicated, stand-alone systems rather than those being accessed remotely and supporting time-sharing systems; those that have simple disk or magnetic tape monitor systems but do not support a large number of input-output units. We mean those that are neither minicomputers on one end of the spectrum nor large number crunchers capable of multiprocessing or multiprogramming at the other end of the spectrum.

In subsequent sections of this paper we will discuss the performance of small computer systems of this type from a number of points of view.

2.1 Speed

A seemingly obvious performance criteria for a computer system is the measurement of the speed with which a particular job can be carried out. The use of this criteria to compare computer systems is fraught with problems stemming from our inability to define a universally useful method of measuring the relevant time periods. To illustrate this difficulty, consider the following relevant time measures: access time, turn around time, and potential for run time optimization. Let us consider each of these time measures in turn.

2.1.1 Access Time

This is the time which elapses between the instant at which the user decides to place a job into the computer and the instant at which the job actually enters the computer. Many users consider this time period to be the most critical. These users are mainly those whose job production is sporadic and possibly tied to their creative output, e.g., the researcher who wants to use the computer to check out an idea he has but does not want to wait a day or more before he can submit his job.

For this type of user, the small dedicated computer system, especially if it is heavily used, has serious disadvantages. He may get to the computer center and find that the only time the computer is free is next week on Wednesday between 3:00 and 3:30 a.m. At that point our user is likely to get discouraged or lock for some other means for getting his job done. The ideal system for this type of person is a time or remote-job-entry system where terminals are accessible at strategic locations throughout the college or university as needed. It should be made clear that such a user might be quite happy to wait overnight for his results, but not to wait several days to gain access to the system.

It is true that there are some small computer systems, such as the PDP 12, which can support remote job entry and/or time-sharing. Use of such systems, however, greatly increases the cost of the basic equipment.

From the point of view of the computer center staff, access time is directly related to program through-put volume. On a small dedicated computer an appreciable amount of time is wasted by the user who signs up for 30 minutes to run a one minute job. Time is also wasted by the user who spends a large fraction of the time that the computer is dedicated to his use deciding what to do next. If a high volume of through-put is needed, the hands-on approach to computer usage becomes unfeasible.

2.1.2 Turn-around Time

Here we are talking about the time elapsed from the instant when the job goes into the system to the instant when the results are printed by the system. This time depends, of course, on the number of jobs being run by the system at that moment, the operating system used, the priority system for running jobs, and the internal speed of the computer used. These times are especially critical in instructional uses of the computer. The number of computer jobs which can be run during a class hour can be the crucial factor, in many cases, in determining the success or failure of the use of the computer in that class. For example, we use our computer in conjunction with a calculus class in such a way that a student introduces a function into the computer which the machine then plots. The student can then change the function or change the parameters in the original function and see the effect on the plot. For the proper interaction between the student and the computer to occur, it is necessary for the response of the computer to be almost instantaneous. A slower response would lead to a loss of interest on the part of many students.

If this measure of computer speed is used, the small dedicated computer system can compete with even the biggest systems. If the machine is dedicated to a job and the problem tackled is not too formidable, the response times of the small systems are more than adequate. To illustrate, the solution of the well known Instant Insanity game typically takes 10-20 seconds of CPU time for a student-written PL/1 program on the 360/50. A similar program written in FORTRAN might take 40 seconds on the 1130. However, it should be clear that on the 1130 the time mentioned is actually the turn-around time for that job. The student user of the 360/50 at a neighboring college in St. Louis would probably wait minutes, if not hours, before he could get his results. In the latter case the turn around time would depend on the usage of the 360 at the time the job was submitted. Turn-around time on the larger system is more unpredictable and this, of course, might be considered a disadvantage of larger systems.

One sees, then, that machines with higher internal speeds may have slower turn-around times for particular jobs. And it is the turn-around time and not the CPU speed that most users are concerned about.

2.1.3 Potential for Run-Time Optimization

Here we are talking not about an elapsed time which is used up in running a program, but rather about the potential for saving run time by using a particular computer system which supports a given set of programming languages. In many instances the best computer system for a given user environment is the one which supports the languages which minimize the users' programming time as well as running time.

A business user will not want to program in standard FORTRAN and, conversely, the numerical problem solver will not want to use COBOL. The user interested in string processing will want a language more suited to him. Considering elapsed time exclusively, it is clear that there is an appreciable saving in programming and running time for the user if the computer system supports a language well suited for his type of problem. Furthermore, if the best language for the job is not clear *a priori*, the user might well expect that the computer center will make available to him a selection of languages representing current programming technology for him to try.

A dramatic illustration of the relationship of language systems used to running time is given in the following example. On the 1130 the solutions to the Instant Insanity game take approximately 2-3 hours when programmed in SL1 (an interpretive language similar to Pl/I written by IBM for the 1130), a day or more when programmed in APL, and 40 seconds when programmed in FORTRAN.

In general, large computer systems support the widest variety of computer languages, but there are a number of small computer systems that are also quite good in this regard. The availability, for example, of FORTRAN, COBOL, SL1, and APL on the 1130 are enough to service the needs of the vast majority of users.

2.2. The Small Computer As A Pedagogical Tool

Since one of the major functions of colleges and universities is the education of students, it seems proper to investigate the efficacy of using the small stand-alone computer as a pedagogical tool.

We have found that a small computer system gives the beginning student a better initial experience with computers than its giant relatives. Many of the liberal arts oriented students who take our beginning computer courses are apprehensive about machines in general and, in particular, about so-called "thinking machines". Such students, we have found, can best be convinced to use computers as part of their learning experience if they are first introduced to them through the use of a machine small enough and accessible enough that the student feels confident he can master its use. In addition, if the machine is operated on an open-shop basis the student receives the reinforcement of obtaining immediate results from the jobs he submits.

As was noted previously, one of the problems of using the small machine is the unavailability of the wider spectrum of software accessible on larger machines. From a pedagogical point of view, it would be desirable to have available to the student, for example, a variety of types of programming languages representing the state of the art in the field. There are available for the 1130, compilers and interpreters for a number of languages even though what is available is only a subset of the language with the same name found on larger computer systems. In recent months string and list-processing languages have also become available for these smaller machines. This software is certainly adequate for the purpose of acquainting students with the rudiments of these languages.

For the computer science student one must be concerned with the adequacy of a computer system for teaching machine and assembler languages. In this pedagogical area the small stand-alone computer offers some real advantages. Its machine and assembler languages are simpler and, therefore, more understandable to the beginning student. The corresponding languages on the bigger machines are much too complex to use, as such, in introductory courses, so the instructor must be content with using only a subset of the languages or a simulator of some simpler lower-level language.

The small dedicated computer with its small turn-around time gives the beginning student the immediate reinforcement he often needs when programming in machine or assembler languages. In as much as it is possible to simulate simpler machines on more complex machines, these advantages can, in principle, be made available on bigger machines, but not necessarily at the same cost per student.

Another advantage of a small computer as a teaching tool results directly from its small core size. This condition brings with it the necessity for optimization of source and object code as a fact of life. Although the same concepts could be introduced on larger machines by using properly sized workspaces as there are in APL, the limitations seem more arbitrary to the student. To give the student the impression that core is limitless and optimizing code is a waste of time is doing a serious disservice to the student. A small system such as the 1130, which allows linking and overlaying of programs, gives even the advanced student the option of writing programs which are reasonably complex without learning sloppy programming habits.

In short, the small computer evidences several pedagogical advantages for the beginning student at a small college. The disadvantages show up when students are advanced enough to need the languages, speed, or space only available on larger machines. In addition, we have assumed that the student generated usage is small enough that we can afford to have hands-on usage with its associated smaller turn-around time. If such usage rises above a critical point it becomes necessary to run the system in batch mode or go to a time shared system. This, of course, obviates some of the advantages described above, as well as adding to the cost of running the system, since it necessitates adding more personnel and equipment.

2.3. Operating System

One of the facts of life at a small academic institution is that the computer center is often a one man operation, especially at the beginning. This means that this one staff member, possibly with the help of a part-time secretary and some student assistants, must do all of the administration of the computer center while at the same time teaching some programming courses. All of the ordering, scheduling, software production, and software and hardware maintenance must be done by this same limited staff.

Therefore, the small college computer center ideally should use a computer system which is as foolproof as possible and easy to maintain. Under the hands-on policy we use at the Lindenwood Colleges it is desirable to be able to have even non-majors in computer science use the machine with a minimum of prior instruction, free of the fear that they will ruin the system.

The small stand alone computer is ideal for this kind of usage. The equipment involved is easy to use and the monitor control language is easy to master. In two years of constant use, our system has rarely been damaged or even shut down because of the incorrect action of a novice user. The major danger to the system from the user comes rather from the advanced student who tries to trick the system into doing something its designers never even dreamed about. But in our experience even in these cases the resulting systems problems are minor. The computer center staffs in charge of larger machines with their much more complicated operating systems and monitor control languages would hesitate to give users this kind of freedom.

2.4. Cost

In the previous discussion several advantages of the small stand-alone system have been mentioned. But in these days of financial pinch it is the relatively low cost of a number of small systems on the market that tend to make these systems so popular. A small college can add a versatile computer system to its facilities for a rental of about $15,000-$20,000 a year (or cheaper if they want to purchase used equipment and pay only for maintenance). At this price they can get a system which can support computer science courses, a variety of administrative uses, and other instructional and research needs of the institution. Such a system does not give one the versatility of an IBM 360 or a CDC 6600 but the difference in cost is formidable.

As is clear to computer center directors but not always clear (at least initially) to college administrators, the cost of a computer center does not stop at the rental or purchase of the computing equipment itself. Peripheral and support equipment (keypunch machines, sorters, cabinets, etc.) and additional staff tend to increase the cost markedly. Most small colleges cannot afford the large staff cost and equipment costs necessary to keep a large computer center operating.

3. REGIONAL COMPUTING CENTERS

One of the suggestions being made to attain the best of both worlds (namely, to have access to large computers at low cost) is the development of Regional Computer Centers. Under such a plan a large computer center, preferably at a university, supplies computer power for surrounding smaller schools through terminals.

It might appear that many schools would find this to be the only way they can have access to a computer system at a price they can afford. But more often a small computer system can be obtained at the same cost. Using a small in-house system as an alternative to participating in a regional computer system has the added administrative advantage that the college which owns or rents the system has overriding control over its use. A remote computer system is usable only at the times and in the manner dictated by the seller of the computer service. The individual institutions must decide, obviously, whether the advantages of a remote system outweigh its disadvantages.

It should be pointed out that there are computer systems which are capable of operating both as a dedicated stand-alone system and as a terminal for remote entry to a larger system. The cost of using the system in both modes is of course higher than either the cost of a terminal or of a stand alone system.

4. SUMMARY

The small stand alone computer system has a number of advantages in the academic environment, the principal one being that such a system evidences considerable versatility at low cost.

On such a system, the turn around time is short, making it ideal for applications where immediate responses are desirable. Such response is especially desirable for pedagogical purposes where the students need the reinforcement of not having to wait for extended periods of time to find out whether or not their program has run successfully.

Smaller machines are easier to use in teaching machine and assembler languages, both because these languages are simpler on the smaller machines and again because of the reinforcement provided the student by immediate response.

The small computer introduces the student naturally to the concepts of source and object code optimization.

The smaller machines are easier to operate and maintain with the necessarily smaller computer center staffs at small schools.

The disadvantages of small machines lie first in the possibly longer access time and lower through-put rate which they provide relative to the bigger systems.

Second, they provide limited space in core for running large jobs quickly. In order for large jobs to be run on the small systems they must be broken up and run in parts, which may greatly increase the run time.

The smaller systems provide less variety of software support for the user. It should be noted, however, that small systems presently on the market provide adequate support for most of the needs of a small college or university.

Our experience leads us to believe that for academic institutions of up to 1200 students the small stand-alone computer system may be the cheapest and most efficient vehicle for bringing computer facilities to the campus.

INSTITUTE FOR EDUCATIONAL COMPUTING

Fred Weingarten
The Claremont Colleges

The Institute for Educational Computing of the Claremont Colleges provides timesharing services via a DECsystem10 (formerly known as a PDP-10) computer to colleges and high schools throughout the greater Los Angeles area. Operating as a non-profit service organization, the IEC is owned by the Claremont University Center, the central corporate body for the group. The hardware link within this network serves as the basis for a cooperative program between IEC and the academic communities it serves.

The network provides on-line service to 13 colleges, including the Claremont group. A research hospital, a junior college and three high schools are also tied into IEC. The nearest user is at our own facility, the furthest is 80 miles away. The terminals, numbering from one to six per campus, are used both for academic and administrative applications, although educational use predominates by far.

This type of use requires that IEC serve as a center for faculty education, software and curriculum development, and as a repository of information about academic applications of computers in all disciplines. In this role, IEC offers its subscribers a wide variety of academic computer services.

Since education of the using community is its primary concern, the IEC provides its subscribers with a wide variety of academic computing services in addition to the sale of computer time. Schools participating in the IEC's timesharing system receive the services of consulting "circuit riders", monthly newsletters, short courses conducted both at the IEC and at off-campus terminal locations, and access to the library of academic software collected and documented by the Institute. The circuit riders are IEC consultants, whose purpose is to provide on-site consulting for the user, to develop small applications programs for him, to teach introductory seminars or programming classes, and to perform minor maintenance and supplying of our terminals. Their time is distributed among these functions as the customer institution sees fit.

The IEC maintains a large library of academic software to support the needs of users in the humanities, social sciences, economics, mathematics, education, psychology, and physical sciences. Programs and applications developed at schools both within the timesharing network and without, are collected, tested, documented, and disseminated by the center. Thus a cooperative exchange of computing expertise, curriculum development, and educational applications is encouraged among the participating institutions.

Development of IEC

Two years ago the Claremont Colleges faced a problem characteristic of one facing many institutions of higher education, that of answering the growing need for on-line computing service for their students and faculty. For the preceeding year, the colleges had had a few terminals to a Caltech DECsystem10 about 25 miles away. Although the service was beset with communication difficulties, enough experience was gained to convince the faculty that the on-line terminal was a valuable educational tool. The need for on-line service could not be met by expansion of existing facilities, which included an IBM 360/40 and an IBM 1130. The 360/40, on the Pomona College campus, was becoming saturated with administrative and other large batch work. The 1130 was used primarily by students and faculty in open-shop mode.

We set several constraints when groping for a mechanism for supplying our computing needs:

1) On-line timesharing service is necessary. It was felt from observing the impact at Dartmouth and other colleges that the educational impact of timesharing was significant. At the same time, research institutions such as the California Institute of Technology, the Massachusetts Institute of Technology, and the Brookings Institute (a private social science research institute) were also using timesharing systems in research environments.

2) The system had to provide reasonable capacity. Based on early assumptions that educational computing meant writing little programs, many people are looking at mini-computer timesharing systems. We rejected that approach because of our feeling that educational computing is not trivial computing.

3) A cooperative approach was necessary to afford the resources we needed. No single college of the Claremont group could, itself, support an on-line computer system. Even if funds could be found, the marginal utility of such a large amount would be greater if spent on a broader spectrum of institutional needs.

4) In a cooperative approach, the distributed costs must be directly and visibly attributable to each category of service. This lesson has been painfully learned by the colleges over their history of cooperation in many fields. Since there is, theoretically at least, a benefit to cooperation, that benefit must always be as apparent as possible to the participating institutions.

5) Extensive user support is necessary. Since the colleges were embarking on an intrinsically expensive venture, even with shared costs, it was deemed necessary to insure that the services were used well and were available to all potential users. We had noted that in many centers, lack of user services was a substantial barrier to many who could make good, effective use of the system. Educational use is even more sensitive to such a barrier.

6) A broader base of users beyond only the Claremont Colleges would help support a faster development of hardware capacity and user support activities. Models for such a broader network existed in the form of the regional centers mentioned in the previous section of this report. On the other hand, no network had started from scratch without a major injection of outside funds. By starting with a larger system we would be able to support a wider variety of applications right away, and therefore have a better chance of starting beyond any threshold which might exist for a system to catch on.

The Institute for Educational Computing was formed as an organization owned jointly by all of the Claremont Colleges. A basic (then PDP-10) DECsystem10 computer was purchased, and the Institute was charged with operating it as an educational resource and selling services back to the Claremont group, other colleges, and other non-profit users in the Los Angeles area. IEC has defined three basic goals it is trying to meet.

A. Provide effective and economic academic computing resources to its customers.

B. Serve as a regional rescurce for faculty consulting and training in computing, and for the development and cataloging of educational software. This service transcends the strictly hard-wired network referred to in (A) above.

C. Serve as an intellectual center for applied computer science, exploring those areas of computer technology of particular importance to educational and research users.

During the first year we had 33 terminals installed plus five dial-up ports attached to the computer. We served twelve colleges, three high schools, two research institutions, and a junior college. The most distant user was eighty miles away. Next year we anticipate from forty to forty-five terminals attached.

With our limited on-line storage, we have developed an off-line file storage procedure which keeps user files on DECtape, small random-access tape files. A simple terminal command system allows the user to call his files off the tape units without worrying about I/O control. Surprisingly to us, this system has run smoothly with very little problems. The quantity of file requests rarely overloads the operator; and users have adapted quickly and patiently to the simple command language and one or two minute delay. In fact, though we have added more disk space, most of it has not gone to permanent user file storage, but mainly to increased public libraries and publically available work space.

The equipment was purchased with the proceeds of a loan which called for interest-only payments during the first year. In this way, we helped alleviate the financial burden of the first year's operation. Later, as use picks up, the hardware will be paid off.

Terminals are sold on the basis of access. That is, users pay for ports into the system rather than time and resources used by them. There are two fundamental advantages of this system to academic users. First, it allows the college to plan its computing budget in advance without the danger of an unpleasant surprise in the form of gross overexpenditures. Also, we have discovered that the control of computer access is an unstable system. That is, small attempts to control access tend to suppress valid student use beyond necessary limits.

Providing terminals and access to a CPU is not what we consider our primary activity, however. On-site consultants work with the users at their installations. The consultants, called "circuit riders", trouble-shoot for the remote users, teach short computing courses or give lectures in regular classes. They also bring problems and complaints back to the operations staff, and so act as ombudsmen for the user. This feedback helps us respond to problems before they become serious. Our consultants have commented that it is a surprisingly different experience sitting 80 miles from the CPU trying to figure out what is happening. Installations providing on-line service can fail for not recognizing that difference.

User manuals and other system documentation are provided, of course. In addition, we publish a monthly newsletter to our users containing information about new software packages and system changes. User complaints of general interest are included in the letter, along with our responses to them. A small journal is also being initiated this summer containing longer articles by our users on their use of the computer in education and research.

Our observation is that software packages, prewritten systems designed for use by the academic community, are becoming increasingly important for educational users. To respond to this new requirement we are building a large library of these applications along with adequate documentation. We are also developing an on-line index to these packages, along with a simple information retrieval system, designed to be useful to the faculty member who is a computer novice, but who is interested in exploring how he could use the machine in his discipline.

There are other IEC activities aimed at stimulating the academic user in the direction of educational computing. A two-week summer training seminar was held to acquaint faculty who were not experienced with computers or in their use, and to start them on software development. The seminar consisted of full-day sessions, mornings spent in the classroom, afternoons in the lab, and used the BASIC language primarily. Another two-week session is being held this summer. These summer programs have been supplemented with lectures by other outside experts in educational computing.

COMPUTERS IN SMALL COLLEGES

Bruce K. Alcorn
National Laboratory for Higher Education*

*The work reported here was conducted while the author was at the Southern Regional Education Board

INTRODUCTION

The impact of computers upon our society is increasing almost daily. This impact, while having reached most segments of our society, is greater in some areas than in others. In the field of higher education the impact is so extensive that the larger institutions could not operate without them.

While computer usage by colleges and universities for the most part started in the larger schools, the pressures for such facilities are rapidly moving down into the smaller ones. They too have computing needs in administrative data processing, in instruction about and with computers, and for some research. In fact, the possession of computing facilities is sometimes used as an element in faculty and student recruiting.

Terms such as "small" and "large" are relative. When referring to colleges in such terms, the description most often refers to student enrollment; however, it can also include variables such as number of levels of degree offerings, facilities, and financial strength. Here, a small college is one whose enrollment is less than 2500 and the highest level of offering is the bachelor's degree.

OPTIONS FOR SMALL COLLEGES

The variety of options of computing facilities for the small college has rapidly increased in the last several years. The following list of options is basically sequential in terms of costs and capability; however, the latitude within each is such that a great deal of overlapping exists.

1. Off-Campus Computers; No Terminals - Colleges have used the computers at other educational institutions, governmental agencies, or commercial organizations by physically transporting the data and/or users to and from the campus. This type of service has usually been thought of as temporary, or specialized, at best.

2. Terminals to Off-Campus Computer - Slow speed inexpensive terminals to a computer at a university or a commercial vendor have been quite successful under the proper situations. Distance between the terminal and the central computer, and the type of use can cause problems.

If the distance exceeds one hundred miles (a rule-of-thumb) the communication costs may exceed the computer charges, especially if regular telephone rates are in effect. Administrative data processing is generally not too practical within this option, as well as other high input/output applications.

3. Cooperative Use of Computer; No Terminals - There are many areas of the country where several small colleges are in very close proximity to each other. In such instances the sharing of one facility can offer better and more flexible service than that provided by the previous option. The major reason why this procedure is not very popular is because in actual practice it is very difficult for such institutions to actually cooperate in major undertakings of any kind.

4. Mini to Small Computer On-Campus - This is probably the most popular option in use today, since it gives the college some computing ability and it also gives them "their own" computer.

There is no standard relative to what a mini-computer is as compared with small, medium and large; however, one reference classifies it in the following manner:

TABLE 1: COMPUTER SYSTEM CLASSIFICATION BY COSTS[1]

Computer Size	Monthly Rental
Mini	under $1,200
Small	$1,200- 5,000
Medium	$5,000- 40,000
Large	$30,000-150,000

The total annual operating costs for systems within this option would probably range from $10,000-$60,000. The hardware would typically range from an 8K system with teletype I/O plus one symbolic and one compiler language on up to a 16K system with disk or tape, card reader and punch, line printer, plus multiple language capability. The smaller of these would be batch systems while others might also have some time-sharing abilities.

5. Cooperative Use of Computer; Terminals - The advantage of this over Option 3 is that each institution has "something" on-campus. Some use of this type is going on; however, at the small college level, some one college usually has the central system and sells "time" to the others. This is then like Option 2 for most of the schools. There are some large-scale examples of this option, notably the Triangle Universities Computation Center (TUCC) in Research Triangle Park, North Carolina and the Middle-Atlantic Educational and Research Center (MERC) in Lancaster, Pennsylvania.

6. On-Campus Computer with Communication Capability - This option ranges in hardware from the high speed remote batch terminal (card reader and punch plus line printer) with no real stand-alone ability (except card to printer), to the so-called "intelligent" terminals to a small computer (such as the higher end of Option 4) with the ability to act as a terminal to a large scale machine. This does provide increased flexibility; however, the cost/benefit situation must be carefully examined.

STATUS OF COMPUTER USAGE AT THE SMALL COLLEGE

To obtain some idea of the status of the situation in the small colleges relative to the use of computers, a look at the study by Hamblen[2] is in order. The information presented here has been abstracted from the inventory report in such a manner as to highlight the small college data. It was not necessarily presented in that report in the same fashion as here.

Table 2 gives the number of institutions of higher education by two enrollment categories and four degree levels. There are 1,413 small colleges as defined here. In other words, 57 percent of all institutions of high education can be classified as small colleges.

TABLE 2: NUMBER OF INSTITUTIONS BY ENROLLMENT[3]
AND HIGHEST DEGREE OFFERED - 1966-67

HIGHEST DEGREE OFFERED

Enrollment	Associate	Bachelor's	Master's	Doctorate	Totals
Below 2500	630*	783*	297	171	1,881
2500 and over	143	48	211	194	596
TOTALS	773	831	508	365	2,477

*Total "Below 2500" and no higher than "Bachelor's"=1,413

Likewise, Table 3 presents population estimates of the number of colleges and universities with at least one computer installed or on order to be in use by the end of 1967. Of the 980 institutions with computers, 301 of them are small colleges. The percentages in Table 4 are computed directly from the data in Tables 2 and 3.

TABLE 3: ESTIMATED NUMBER OF INSTITUTIONS WITH COMPUTERS[4]
BY ENROLLMENT AND HIGHEST DEGREE OFFERED - 1966-67

HIGHEST DEGREE OFFERED

Enrollment	Associate	Bachelor's	Master's	Doctorate	Totals
Below 2500	148*	153*	92	78	471
2500 and over	105	36	182	186	509
TOTALS	253	189	274	264	980

*Total "Below 2500" and no higher than "Bachelor's"=301

TABLE 4: ESTIMATED PERCENT OF INSTITUTIONS WITH COMPUTERS
BY ENROLLMENT AND HIGHEST DEGREE OFFERED - 1966-67

HIGHEST DEGREE OFFERED

Enrollment	Associate	Bachelor's	Master's	Doctorate	Totals
Below 2500	23%	20%*	32%	46%	25%
2500 and over	73%	75%	86%	96%	85%
TOTALS	33%	23%	54%	72%	40%

*Total "Below 2500" and no higher than "Bachelor's"=21%

The computer hardware actually reported[5] at the small colleges is listed in Table 5, with estimates of expenditures for computer facilities in Table 6.

TABLE 5: COMPUTER REPORTED IN USE OR TO BE IN USE BY THE END OF 1967 AT INSTITUTIONS WITH ENROLLMENTS LESS THAN 2500 AND OFFERING AT MOST A BACHELOR'S DEGREE.
(1125 out of 1413 reporting)[6]

Type	No.	Type	No.	Type	No.
BUR E 103	1	HON 200	3	IBM 1130	48
BUR 101	1	HON MARK XI	1	IBM 1401	36
BUR 205	3	HON 2000	1	IBM 1440	1
CDC G 15	2	HON 3000	1	IBM 1620	49
CDC LPG30	1	PDS 1020	1	IBM 360	1
CLY DE 60	1	UNI ATHENA	1	IBM 360/20	9
DEC PDP 8	1	UNI 1004	1	IBM 360/30	4
GEC 115	1	UNI 9300	1		

TABLE 6: ESTIMATES OF 1966-67 EXPENDITURES FOR COMPUTERS BY ENROLLMENT AND HIGHEST DEGREE OFFERED IN MILLIONS OF DOLLARS[7]

HIGHEST DEGREE OFFERED

Enrollment	Associate	Bachelor's	Master's	Doctorate	Totals
Below 2500	8.5	5.0	6.2	18.6	38.3
2500 and over	13.0	3.7	20.5	145.6	182.8
TOTALS	21.5	8.7	26.7	164.2	221.1

*Total "Below 2500" and no higher than "Bachelor's"=13.5

It is interesting to note that according to this study, the small colleges comprise 57 percent of all institutions of higher education, make up 31 percent of the colleges with computers, and are responsible for 6 percent of the funds expended for computers by colleges and universities.

AN EXPERIMENT

Description:

In 1968 an experiment was initiated in order to gather some data pertinent to the use of computers for instructional purposes in small colleges.[8] Three different ways of options for obtaining computing facilities are involved--terminals to off-campus computers, small computers on campus, and cooperative use of a computer without terminals (Cluster of Colleges or Group).

Figure 1 identifies the institutions participating, shows their locations and indicates the option in which each is involved. Some descriptive information about each college is given in Figure 2, and a listing of the hardware at, or used by, each is presented in Table 7. This latter table, as well as all the following data represent the status as of May 31, 1970.[9] This experiment will be essentially completed at the end of August 1971 with a final report to be published by the end of that year.

The Role of SREB:

The staff of the Computer Science Project of SREB are involved in a variety of activities contributing to the experiment, in addition to collecting the usual data and administering the project. These activities can be identified in three groups--Evaluation, Materials, and Services.

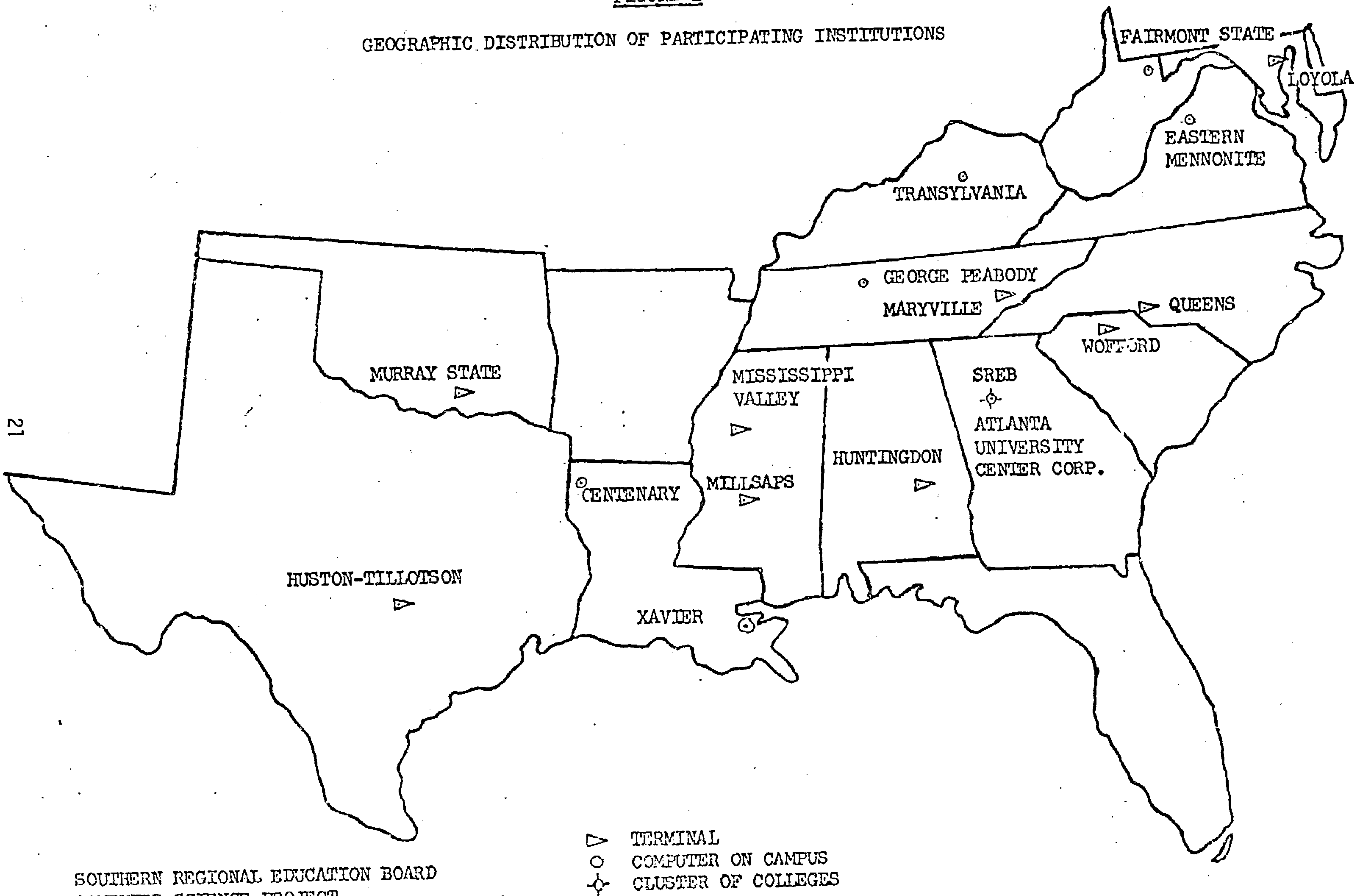
1. Evaluation - A case study is being written for each institution with the purpose of allowing others to see what kind of colleges did what with how much. In addition, an attitude questionnaire was designed (to measure attitudes toward computers in general) and is being administered to the students, faculty and administrators of the participating schools to see if attitudes toward computers have changed during the experiment.

2. Materials - Several items were, or are in the process of being produced to assist the colleges. They include an annotated bibliography for introductory courses in computer education, a prototype users' manual for a small college computer center, a listing of the time-sharing services available in the SREB region, a collection of 135 problems useful in teaching programming, computer science, mathematics, physics, business, economics, and statistics, a short course to provide an introduction to IBM 1130 FORTRAN programming, and an instrument for determining machine independent measures relative to courses utilizing the computer.

In addition, faculty at four of the colleges received one-year grants from the NSF for computer related curriculum development in chemistry, economics, natural science, numerical methods and physics. The materials developed were disseminated and demonstrated to and critiqued by faculty from all the colleges in a series of two workshops.

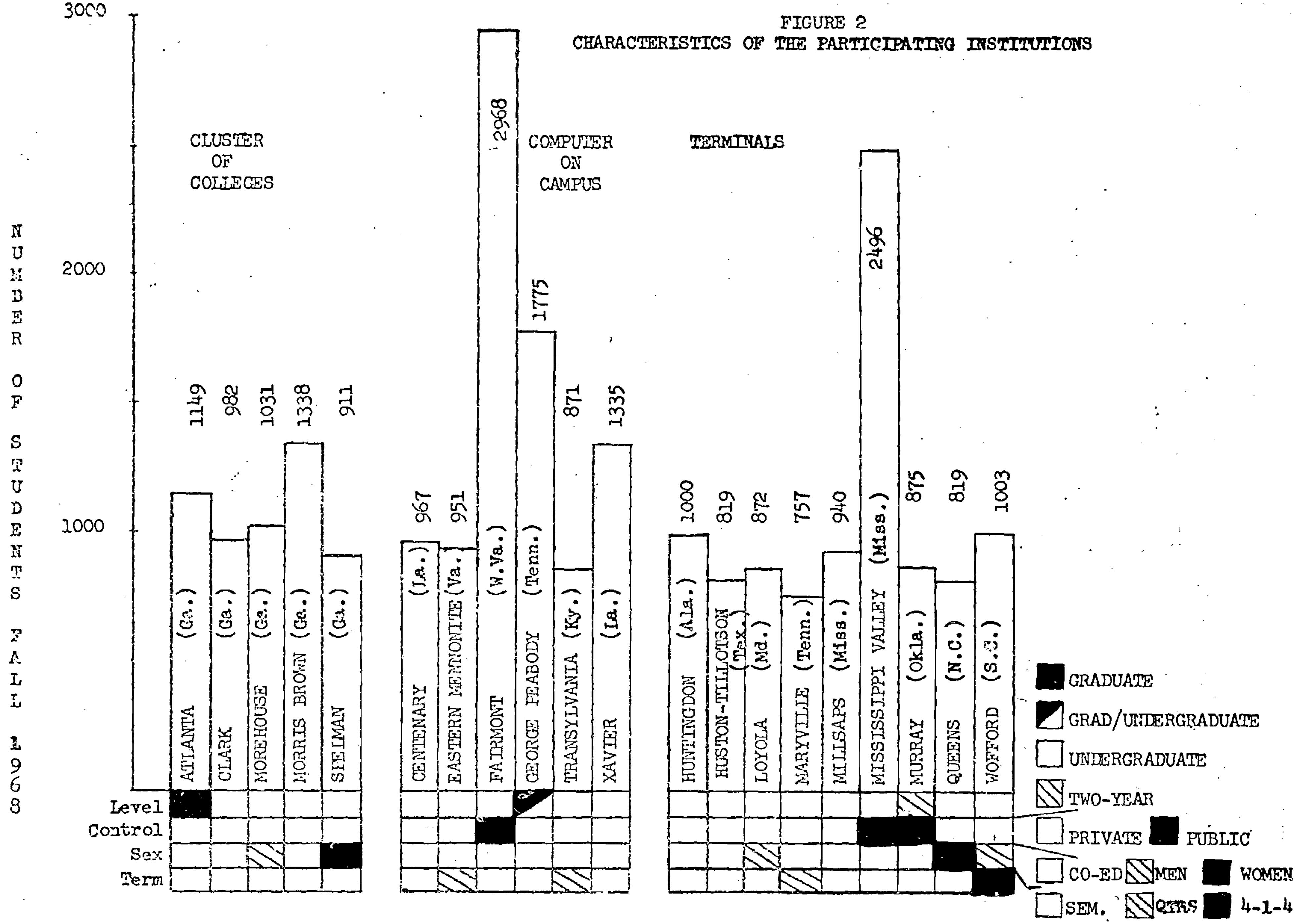
FIGURE 1

GEOGRAPHIC DISTRIBUTION OF PARTICIPATING INSTITUTIONS



SOUTHERN REGIONAL EDUCATION BOARD
COMPUTER SCIENCE PROJECT
NSF COMPUTER EXPERIMENT
NSF Grants GJ 269-275, 277-280, 330, 404-6, 417, 481

FIGURE 2
CHARACTERISTICS OF THE PARTICIPATING INSTITUTIONS



NSF Regional Computer Experiment
Computer Science Project
Southern Regional Education Board

TABLE 7

FACILITIES STATUS AS OF JUNE 1, 1970

INSTITUTION	ON-CAMPUS HARDWARE 1969-70 % R & I* USE	MOS. OF USE	UNIT	OFF-CAMPUS HARDWARE MODE OF USE	UNIT	INSTITUTION/COMPANY LOCATION OF COMPUTER
Loyola	100	12	ASR-33(1)	IN-WATS	GE-265	Call-A-Computer, Inc. Long Island, N.Y.
Maryville	100	12	ASR-33(2)	IN-WATS	GE-265	Call-A-Computer, Inc. Los Angeles, Calif.
Wofford	100	12 1	ASR-33(3) H-P7200A PLOTTER	IN-WATS	GE-265	Call-A-Computer, Inc. Los Angeles, Calif.
Huntingdon	100	9	IBM 2741(4)	LEASED LINE	IBM 360/50	Auburn Univ, Auburn, Ala.
			ASR-35(4)	IN-WATS	GE-265	United Computing Systems Kansas City, Missouri
Millsaps	100	9	ASR-33(5)	IN-WATS	GE-265	Call-A-Computer, Inc. Los Angeles, California
			ASR-33(5)	IN-WATS	GE-265	United Computing Systems Kansas City, Missouri
Mississippi Valley State College	100	10 12	IBM 1050 IBM 360/20(6)	LEASED LINE	IBM 360/40	Mississippi Research and Development Center Jackson, Mississippi
Murray State	100	12	IBM 1050	LEASED LINE	(8) IBM 360/50	University of Oklahoma Norman, Oklahoma
Huston-Tillotson	100	8	ASR-33(7)	LOCAL TELEPHONE	CDC-6600	University of Texas Austin, Texas
Queens College	100 97	12 6	ASR-33 IBM 1050	IN-WATS FX-LINE	IBM 360/75	Triangle Universities Computation Center Research Triangle Park, N.C.

(1) Two units capable of on-line use for the whole period. A third unit, for off-line tape preparation only, added from December-May. Some free time available on an IBM 1130 and a UNIVAC 1108 at local industries.
(2) A second unit added from January-May, primarily for off-line tape preparation.
(3) A second unit added from November-May for off-line tape preparation only.
(4) Auburn University services used September-December; United Computing Systems from January-May.
(5) Two units capable of on-line use. IBM 1130 at Jackson State College utilized as part of several courses. CAC used from September-December; United Computing Systems from January-May.
(6) Used for administrative data processing only.
(7) A second unit added in November.
(8) Prior to December 1, 1969, the central computer was an IBM 360/40.

*Research and Instruction

INSTITUTION	1969-70 % R & I USE	MOS. OF USE	ON-CAMPUS UNIT	CORE STORAGE	DISK STORAGE	CARD READ	SPEED PUNCH	LINE PRINTER SPEED	MISC.
Peabody	86	12	IBM-1130(1)	8K	1536K	300	60	340 l/m.	
Xavier	54	12	IBM-1130	8K	512K	300	60	80 l/m.	
Centenary	74	12	IBM-1130	8K	512K	400	60	80 l/m.	Storage Access Channel II
Eastern Mennonite	57	12	IBM-1130	8K	512K	300	60	80 l/m.	
Fairmont State	60	12	IBM-1130(2)	8K	512K	300	60	80 l/m.	
Transylvania	70	10	IBM-1130(3)	8K	512K	300	60	80 l/m.	
Atlanta University Center Corporation	100	12 12	IBM-1130 IBM-1401(4)	16K 12K	1536K 3000K	340	60	340 l/m. 600 l/m.	1232 Optical Page Reader 1627 CAL-COMP Plotter

(1) Also available is a HP2114A on campus, plus an XDS SIGMA 7 at Vanderbilt University.
(2) During the last 2.5 months of the reporting period an on-campus ASR-33 was connected to an IBM 360/75 at West Virginia University.
(3) Also available is the IBM 360/65 at the University of Kentucky. An IBM 2741 terminal to the University of Kentucky was used some as part of a German course.
(4) Used for administrative data processing only.

3. Services - Arrangements were made for a common library of programs for those terminal users using the same time-sharing supplier. Each institution was visited several times by the project staff and seven group meetings were held with the principal investigators. These meetings involved a variety of topics from installation and management problems to curriculum development project reporting.

Two-Year Statistical Results:

Figures 3-9 give some indication of the extent of the activities during these two years. Of the twenty institutions involved, 13 have completed two years of computer activity and the remaining seven, one year. The figures show this with pairs of columns for each type of facility. The one labeled "First" presents the first year data for all participating institutions regardless of whether their first year was 1968-69 or 1969-70. "Second" presents the second year data only and involves only those institutions which started with the 1968-69 academic year.

Figure 3 depicts the equipment utilization in terms of the average number of console hours used per month. This is only an approximation since console hours are not always measured in the same way. It must also be remembered that for the "Group" this is the average for the entire group and not the average for each institution in the group. The (N=1) refers to one group of five institutions. It should be noted that the colleges with the "Small Computer" facilities also utilized them for administrative data processing.

Figure 4 gives estimates of the percentages of the total student bodies (registrations) using the computer facilities for instructional or research purposes. Figure 5 is analogous, but for faculty participation. The data in parentheses and the portions of the "First" columns below the broken line are the first year data for those institutions having completed their second year. In other words, the first year (1968-69) data is separated from the 1969-70 first year data. For example the three colleges (N=3) which have completed two years, had 16 percent of their faculty participating during 1968-69, their first year and 19 percent during their second year, 1969-70.

Figure 6 shows the percentage of total facilities utilization for instruction, research and administration. A separate IBM 1401 center is maintained at the Atlanta University Center (Group) for administrative data processing. A separate IBM 360/20 is similarly used at Mississippi Valley State College. Administrative use of small computers (IBM 1130's) ranged as high as 50 percent of the total utilization at one campus during the first year and 46 percent at the same campus the next year.

Although approximately four times as much was expended to maintain the small computer facilities as was spent on the terminal facilities (see Figure 7), the average expenditure for instruction and research use per registered student was only twice as much during the first year and one and four-tenths as much during the second year (see Figure 8).

Since the data shown in this report are partial (not all institutions have had computer facilities for two years), and what is even more critical, the data is for the initial start-up periods, interpretation should be kept in this framework and attempts should not be made to project these data as being indications of later years of operation.

Figure 9 presents the average cost per participating student in each of the major academic disciplines or groups within and for each type of facility. The seemingly high figures for the first year for the "Group" ($230) can be partially explained by the intensive use of computers by students in programming, computer science and social science. If these costs appear high, even in their present context, the reader should bear in mind that few, if any, other aspects of college instruction are allocated on this basis. Usually such cost as library operations are computed as average costs per student enrolled and not on the basis of usage.

Problems:

The problems related to the establishment and operation of a computing facility in a small college are basically the same type as in a large university, only magnified because of the inexperience. The start-up pains are of shorter duration however, than they were ten years ago.

The usual hardware and software "bugs" are still around, including difficulties with vendor service. The colleges often started out with, or grew into, a shortage of space, staff, card punches or terminals and computer related curriculum materials. This was followed by pressures for plotters, faster printers, additional software packages and better scheduling priorities.

A very consistent problem was that of a lack of faculty with computer experience, and heavy teaching loads which made it difficult if not impossible at times for inexperienced faculty members to overcome this deficiency.

FIGURE 3

AVERAGE CONSOLE HOURS PER MONTH

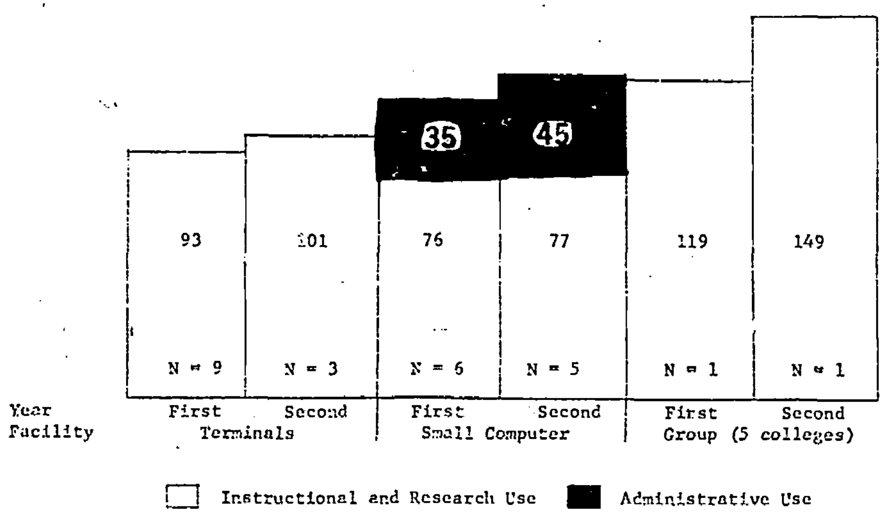


FIGURE 4

ESTIMATED PERCENT OF STUDENTS PARTICIPATING

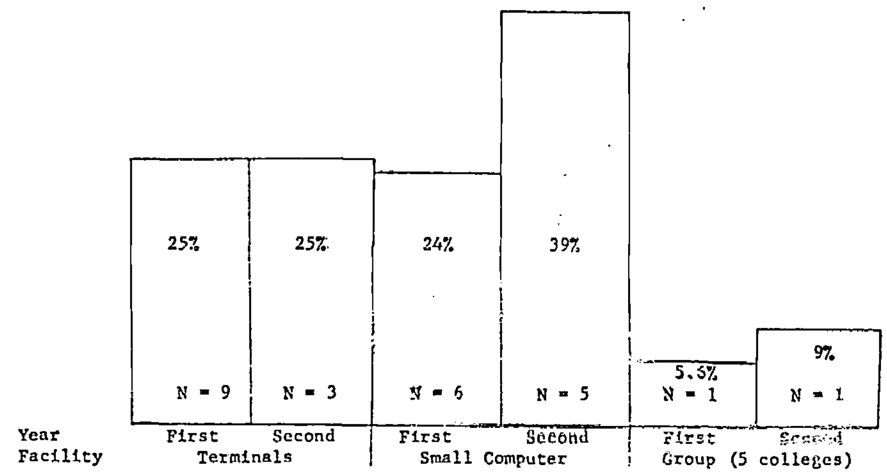
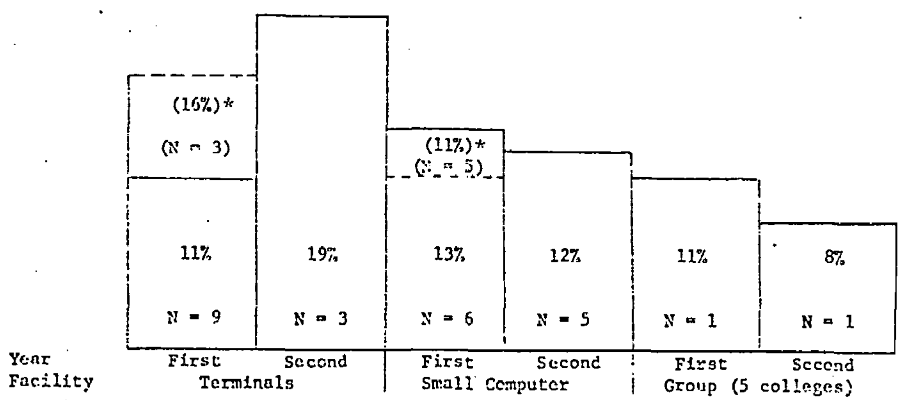


FIGURE 5

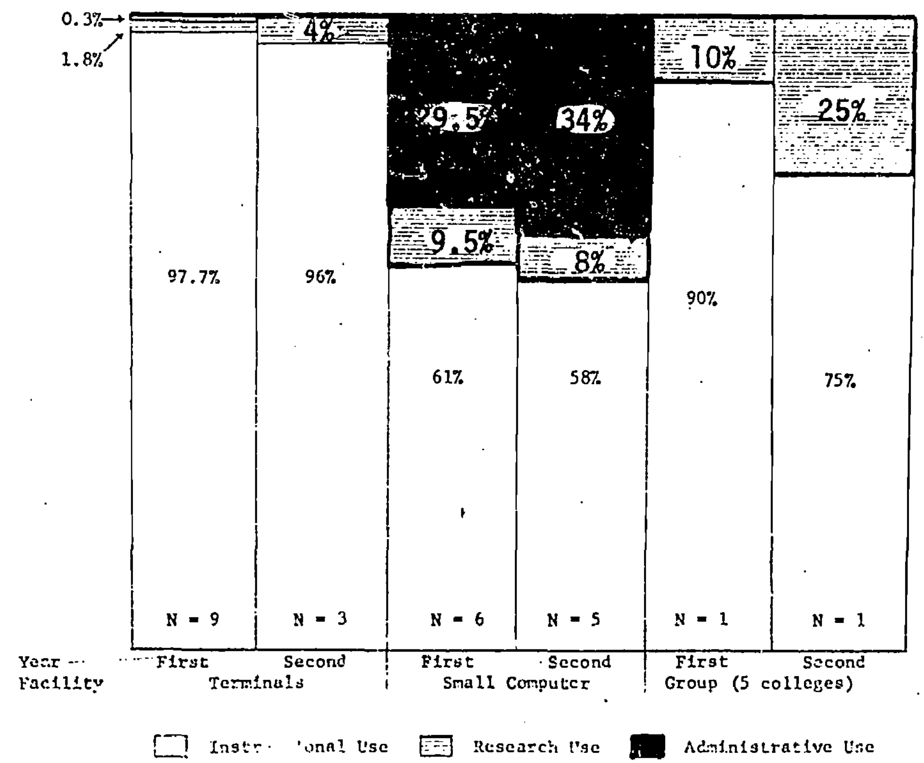
PERCENT OF FACULTY PARTICIPATING



*------ The percent of faculty participation for the first year for the same
( ) institutions for which second year data has been reported.

FIGURE 6

TYPE OF USE



NSF Regional Computing Experiment
Computer Sciences Project
Southern Regional Education Board

FIGURE 7

AVERAGE TOTAL EXPENDITURES
(Instruction-Research-Administration)

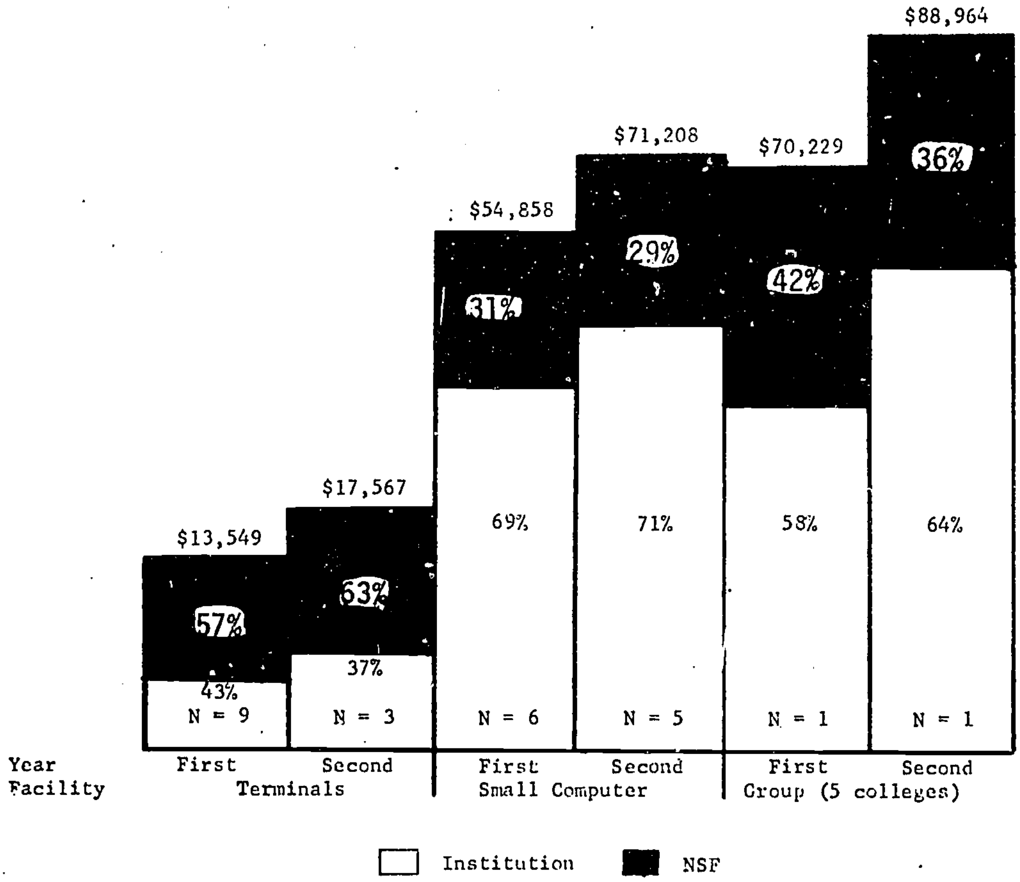
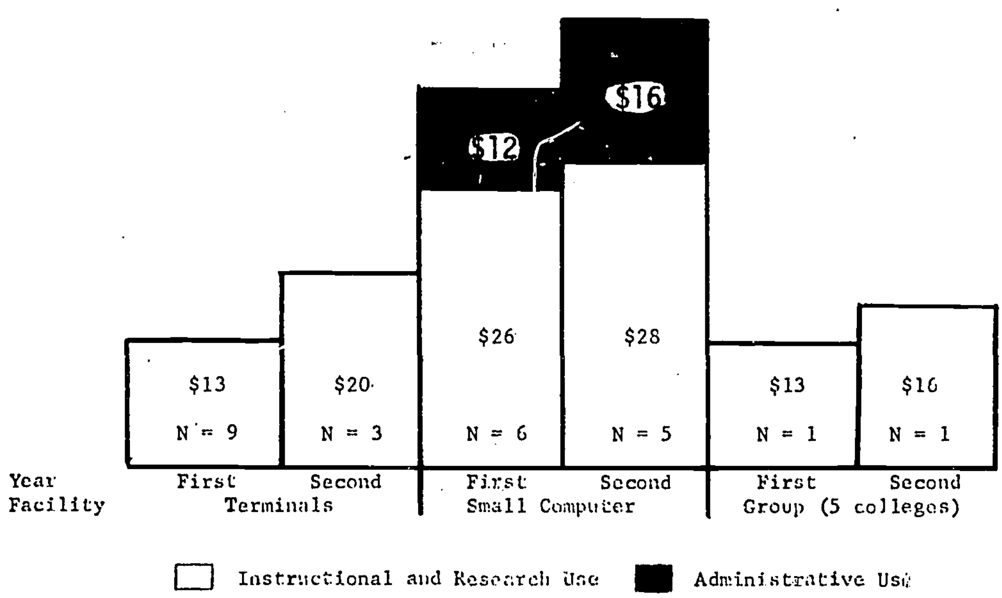


FIGURE 8

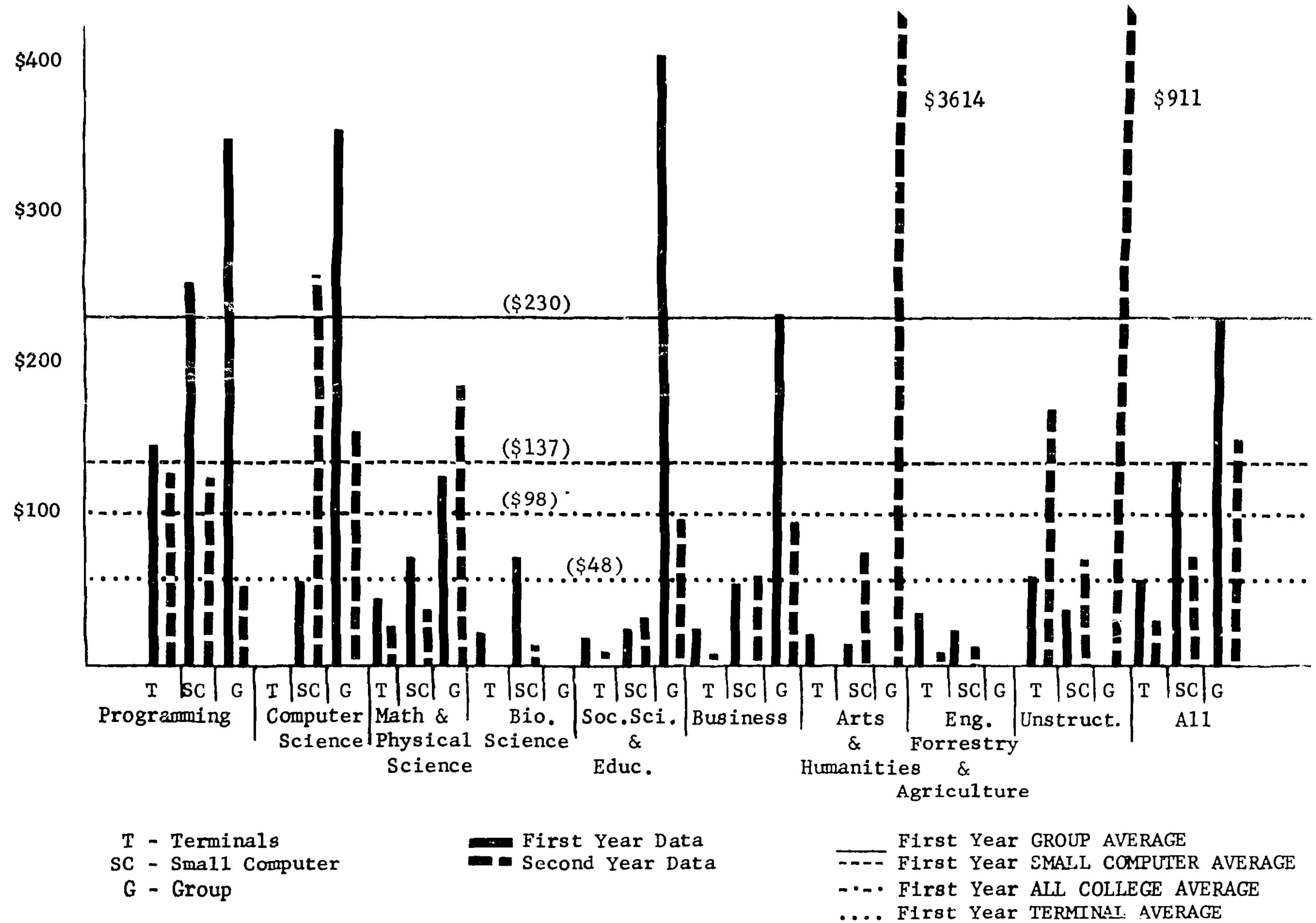
AVERAGE EXPENDITURE PER REGISTERED STUDENT



NSF Regional Computing Experiment
Computer Sciences Project
Southern Regional Education Board

FIGURE 9

AVERAGE COSTS PER STUDENT PARTICIPATING FOR INSTRUCTIONAL USE



NSF Regional Computing Experiment
Computer Sciences Project
Southern Regional Education Board

Temporary Summary:

Even in this short period, there has been enough evidence to draw some conclusions, even if not final. In most cases the project has been successful from the college standpoint. In fact, some have admitted that they would be lost without computer facilities.

Every one of the twelve institutions which is no longer receiving NSF support are now operating on their own at the same level or even greater. Some even feel that "their computer" is a positive element in recruiting both students and faculty.

The beginners do need help in most areas, but especially in getting the faculty actively involved. The success of such facilities is strongly related to the leadership, enthusiasm, and attitudes of those really in charge of them; that is, the personnel making the important decisions.

A Caution:

The reader should be cautious in interpreting the data related to this experiment. The emphasis is placed upon utilization of the different types of computer facilities for instructional purposes and does not discuss research and administrative uses, real or potential. The reader should keep in mind the following as he interprets this report:

1. The initial thirteen participating institutions had little or no computer use prior to July 1, 1968, and the latter seven had little or none prior to July 1, 1969.

2. The "first year" of computer use covered by this report varies from six to 11 months (see Table 7). The "second year" is a full 12-month period.

3. The average total computer use in hours per month during the second year ranged from 85 to 229 or from 24 percent to 65 percent of capacity based upon a maximum (conservative) two shift operation (352 hrs./mo.).

4. The characteristics of the institutions differ considerably (see Figure 2).

Because of the above, costs per participating student are still inflated as compared with similar costs at centers which are several years old. However, the total costs do represent good measures of "start-up" costs for the different types of facilities included. Second year total costs increased 20-25 percent while costs per participating student showed a significant decrease. As computer use becomes greater the operation of the facilities becomes more efficient and unit costs will tend to level. This leveling can be expected when the facilities approach two full shifts (352 hrs/mo) of usage. This will probably occur during the third or fourth years of operation.

Footnotes

[1]John W. Hamblen, "Central Computer Center Organization and Computer Systems Options for Institutions of Higher Education," prepared for Proceedings of Conference on Computers in Instruction: Their Future in Higher Education, RAND Corporation, October 1-3, 1970.

[2]John W. Hamblen, Inventory of Computers in U.S. Higher Education, 1966-67, (Washington: National Science Foundation, 1970).

[3]Ibid, I-5

[4]Ibid, II-3

[5]Of the 1413 institutions in this group, 1125 reported.

[6]John W. Hamblen, "Computing Facilities in the Small Institutions: 1966-67," Proceedings of a Conference on Computers in the Undergraduate Curricula. (Iowa City: Center for Conference and Institutes, University of Iowa), 11.19 and 11.20.

[7]John W. Hamblen, Inventory of Computers, 1966-67., Ibid, III-2.

[8]"Experiment on Ways of Supplying Computer Facilities to Small Colleges for Instructional Uses" is supported by NSF Grants 269-275, 277-280, 330, 404-406, 417, 481 and is being conducted by the Computer Sciences Project of the Southern Regional Education Board (SREB). The Project Director is John W. Hamblen and the Associate Project Director is Bruce K. Alcorn.

[9]Another related experiment was initiated by SREB during the summer of 1970. It involves yet another of the options, mini computers, and is called "An Experiment on Utilizing Mini and Very Small Computers for Instructional Uses," NSF Grants 1072 and 1111. This project has a duration of three years, involves ten colleges in seven states (including Maryville College), and includes products of three different manufacturers.

COMMENTS ON SOURCES AND TYPES OF SERVICE

Whitney Johnson
Virginia Council of Higher Education

I'd like to ask for a show of hands: How many of you represent state-supported institutions? How many represent private institutions? Looks like just about half and half. Okay. I think that we all know, and it probably is appropriate to re-emphasize, that the computing needs of any educational institution fall into several categories. We may narrow these down to three, roughly, as being the instructional support area, the research support area, and the administrative support area. (keeping grades, student records, registration, and what-have-you). Of course some would have to add extensions to public service and other areas, but you can usually break those down into components that fit into one of these three. Whether you have a small institution with a hundred students, or an institution with thirty or forty or fifty thousand students, you have all of these components having a need for computing at your institution. Now some of you are only interested in one of them. But someone at your institution also has to be interested in the others. As our keynote speaker indicated, there are cases where you simply can't afford to get involved with the computer in some of the administrative tasks, and I think we should recognize that. We, as the computer people on campus, need to look at projects to make a decision, or to help our administration make a decision as to whether it is cost-effective to do it this way, or should we really just say, "Keep doing it by hand. It's the best way."?

When you get involved with a statewide system of higher education some way, you find that the people begin to look at the money you spend in a little different manner. They begin to say, "Well now, let's see. We don't care, really, whether it costs you more from your budget to do it this particular way. In the end result, it costs the state and the taxpayer less to do it this way.". So very frequently the concern is not with the cost on an individual campus. The concern turns out to be with the cost to the taxpayer and the state system. So you have movements around the country towards setting up large, massive central systems to serve the needs of the state.

It was not too long ago that right here in Georgia this was a heavily debated issue in the legislature. There was a move here to put all computing within the state--that is all computing for the various state agencies and state-supported educational institutions--into one massive, centralized computing network. That included all of the systems in the colleges and universities. That included all the people, too. There would not be any Computer Center Directors reporting to college presidents. They were going to report to the computing czar at the state capitol. Well, fortunately--I think--there was enough discontent registered that when it finally got through the legislature, the educational institutions had been exempted from this massive computer-czar arrangement. However, I think that these institutions see the handwriting on the wall and recognize that they'd better clean up their own house, so to speak, in terms of getting the dollar's worth for the taxpayer within their own system of higher education.

In other places around the country, we see regional computing centers being developed. That's what we are trying to do in Virginia, for example. They are doing it in other states. They are encouraging the smaller institutions, located reasonable near, geographically, to major educational institutions, to go to the major institutions for their primary computer support. This essentially says, then, that some of the smaller colleges may only have a card-reader-printer-type real, batch station. Others may have an IBM 1130 with a synchronous communications hookup on it, going into the major computer center. The latter type of operation, in my opinion, has several advantages to the smaller institutions. It doesn't say that they can't have some small computing capability to give the student some hands-on capability, but it does give them access to have major computing concepts, the major computing languages, the wide variety of languages, the wide variety of support at the major university nearby.

Now in Virginia we made it a specific point to include the private institutions as well as the state institutions in our plan. We feel that by hooking up to one of the regional centers, you can get a lot more computer power for your dollar than you really have to pay for that that you actually use.

I think we see generally around the country a tendency to move more towards what you might refer to as a "hard money" policy in university computing centers. That is, the open access idea of the library is beginning to be moved aside in terms of, "If you've got money, you can come and buy computer time. If you haven't got money, we'll let the computer sit idle.". To some of us that is a problem because we think computer cycles go to waste. But the user must recognize that it costs money to use it. Hard money is one way to do this.

I would like to make reference to our keynote speaker's talk again. He said that computer costs represent about half of library costs. I wasn't quite sure where he had gotten that. I hoped he would be here, but I saw him leaving at break time. As I recall, the President's Advisory Committee suggested that computer support ought to be about equal to library support, and that library support ought to be, as Dr. Reveley indicated, about five percent. I did a little survey of the state higher-education agencies last December, asking how much had been spent in each state for education and general expense. (now this, of course, runs everything together, but there is a question of definition,). How much had been spent for computing support, and how many students, count and full-time-equivalent, were involved. Of course, the information I got back varied, depending on the responsibility of the individual who received the questionnaire. In some cases I got figures for the whole state system; in some cases I got figures for part of a system; in some cases I got numbers for one individual institution.

I think the interesting part is that approximately three percent of the total educational general expense is going toward support of computers. Some states were as high as seven percent; some states were below one percent. Overall, it was around three percent. This also shakes out to be about forty-five dollars per head-count student, or fifty-five dollars per full-time-equivalent. Now naturally there are more heads than there are full-time equivalents because you get part-time students added in. I don't know what all this means. I don't want to imply that this is enough, or anything like that; but it gives you a picture of what is being spent around the country.

Now, we have in Virginia one regional computer center operation that I would consider fairly successful, the Center at the College of William and Mary, working on a System/360 Model 50. There are twelve institutions, both public and private, tied into that center. The community college system is just completing what they call a student information system, with all of their student files for those institutions in the community college system that they are bringing up now--they're not doing them all at once, but they have about five--all of their files and student records are going to be maintained at the regional computing center. There are parallel operations at two institutions this spring that have run very nicely. We think this is a good illustration of what some cooperative effort can generate.

Another illustration from this same regional center, we have an institution that is actually larger in student body than the College of William and Mary that had a 1620 and wanted to upgrade it. We encouraged them to go to something that would tie into the regional center. Then, of course, they wanted people to write all kinds of new programs, and we said, "Look, why don't you just go over to the regional center and talk to them and see what's there?". When they finally did this, the report came back to us, "They've already got programmed ninety percent of what we need.". And so we have this institution that is picking up packages that were already running at William and Mary, and doing their accounting, their student records, and all of these kinds of things that William and Mary was already doing, so there was no reason why this other institution couldn't use them if they wanted to. I think most of you would have found this true had you looked at these types of alternatives. You will find that the university nearby does have a fairly extensive package of administrative data-processing programs. Though you may have to give a little bit on what you have been doing, you can usually use those programs very, very effectively at almost no programming cost. And that is very important to you, it seems to me.

We are more and more in the state systems running into state data-processing systems or state data-processing agencies that are not involved in higher education, and these concern me. I personally believe that coordination of computing in higher education should be maintained within the family of higher education. Of course this is where I differ with the director of the Division of Automated Data Processing in Virginia.

DISCUSSANT COMMENTS ON SOURCES AND TYPES OF SERVICE

Glenn Ingram
Washington State University

When Jerry Engel asked me to serve as a discussant for this session, I said I'd be pleased to, but on second thought, asked how I should prepare for the job. His advice was, "Have your sense of humor, be prepared to fill in for Preston Hammer if he doesn't make it, and if Gordon Sherman doesn't show up, you can talk about NSF and ACM."

Reflecting on these three points, I realized with some sadness that I was ill-prepared for the role. First, as a Computing Center Director, I don't have a sense of humor. Second, as an old Giant fan, I've had a recurring dream of being called to pinch-hit for Willie Mays, but I didn't have enough time to grow a beard or acquire the other attributes necessary to pinch-hit for Preston. And third, I believe Milton had a somewhat easier task in justifying the ways of God to man than Gordon had in justifying the ways of NSF to small colleges.

Since my task wasn't rigidly defined, I'll feel free to comment on this morning's speakers as well as those of this session. Jerry led off with a description of the duties of a computing center Director in a small college, noting that you taught computer science classes, probably served as assistant dean, perhaps as registrar, and quite likely coached football on the side. This reminds me that in the old Jewish Bible, the Books of Kings, Jeremiah, Ezekiel and Isaiah were arranged in that order, with the rationale that the Books of Kings ended in desolation, Jeremiah was all desolation, Ezekiel begins in desolation and ends in consolation, and Isaiah is all consolation. While the day of Isaiah is not upon us, there are some rays of hope.

One of Preston Hammer's subtle comments provided one ray. Of course, he had wounded me earlier with the remark that anyone who could remember the CPC was too old to be here (I started computing with a 402 and 602A, and thought a CPC was really a computer!). But his word of consolation came in his definition of a valid computer application: if you really know how to do a job, turn it over to the computer. Be reassured - no one will turn a Computing Center Director's job over to a machine!

Gordon Sherman explained NSF's primary mission, and in a response to a question, indicated that there was little prospect for direct help to small colleges. As I indicated earlier, I can feel sympathy for Gordon's position, because I found myself in similar spots a number of times during my years at NSF. But if we accept the view that the Sabbath was made for man, and not man for the Sabbath, then a concomitant view should be that federal agencies were created to help institutions of higher education, and not the converse.

The point of this is that although we've all taken parts of our educational responsibilities very seriously, we've all neglected one significant part. We have worked hard to introduce the computer to our faculty colleagues and our students, and to explain it to our administrations. But how many of us have tried to educate taxpayers or legislators to the importance of computing? We may let off some steam in talking to Gordon as a representative of NSF, but he doesn't set policy - some ex-Pfc. Wintergreen in the Office of Management and Budget does that. You all have Congressmen and Senators, and you have an educational challenge there.

The presence of Dr. Reveley at a meeting of this sort is significant. Just the fact that, as a college president, he had faced the question of whether his college needed a computer, and had reached an affirmative conclusion, is important. But obviously, he has investigated issues related to computing, and its role in a liberal arts college. He had studied costs for computing centers nationally, and compared them with library costs. This is a frequent comparison, and it has some hazards that I'll return to later.

Dr. Reveley spoke of the exchange of memoranda with his key faculty on the subject of computing, and at one point came to the surprising conclusion that the Computing Center Director can boss everyone on campus. Jerry Engel must write eloquent memoranda. By contrast, I recall the case of an exchange of memoranda in which the president criticized one originating in the Computing Center as laden with jargon and cliches. The president received another memorandum from the Computing Center Director noting that his job required that he put his shoulder to the wheel, keep his nose to the grindstone, his eye to the future, his ear to the ground, and the wolf from the door, and he didn't have time to develop a new writing style.

I've filibustered long enough, and will turn to this afternoon's session. Aaron Konstom gave a good framework for the typical small college center, observing that it was essentially a one-man operation, that it had to be cost-effective, with cost as the over-riding consideration, and that a Director typically adopted a minimizing principle of operation: make as few people angry as possible. He also noted educational advantages of a small computer: students are aware of the finiteness of resources, and learn criteria for the quality of a program - reduce core size and optimize run time; further, there is an opportunity for hands-on experience. I believe the first of these is a real advantage, and it is true that big - computer people have a certain type of provincialism that leads them to underestimate how much can be done on a small computer. The second point - hands-on experience - should be questioned, I believe. The fundamental question is hands-on what? Clearly, anyone who submits a job to a computer has had his hands on something - deck of cards, keypunch, terminal, etc. But how much is learning advanced by actually operating a computer?

This leads into the next presentation by Fred Weingarten. Fred outlined a half-dozen approaches to regional cooperative computer uses, and listed the five criteria on which his own center chose its equipment and organized its operation. He noted that in his operation, when students or faculty had a choice between operating a stand-alone computer or a terminal, terminals were much more popular, and stand-alone use withered. This may not be a universal phenomena, and many of us can't afford the luxury of providing the choice. However, it does provide a commentary on hands-on-a-computer experience, and suggests that alternatives may be viable.

Two other comments by Fred merit attention. He said that one of the criteria for his center was, "Computing service alone isn't enough." We all believe this, and I don't know what the implications are for a basically one-man operation, but we are back to an educational problem. He also called attention to matters of financing, billing and accounting. Some of you have said that you don't bill for computing services, but I doubt that you can continue to avoid it. Personally, I don't believe the meek will inherit the earth - I believe auditors will - and for this reason as well as for internal justification, adequate accounting seems necessary and inevitable.

Whitney Johnson enlarged the scope of discussion of cooperative efforts to the statewide level, and in doing so, forces us to examine some potential problems. This comment may seem to apply only to state-supported institutions, but I believe there will be a very strong trend to statewide networks that may include private colleges as well. The burden this will place on private colleges is the examination of the alternative way of securing computing services. And one cannot dismiss it with the argument that communications costs are high if the distances are large. To return to one of Aaron's points, it is the total cost that must be considered, and computing costs may well be low enough to offset the communications hurdle. Nearly all of us who have been involved in regional computing projects have found that participating colleges have overestimated the amount they would spend on computing services.

Whitney noted that those individuals who propose state-wide networks tend to view this from the standpoint of total cost for the system, and suggested that the global cost optimization might not coincide with optimizing costs on a local basis. He also pointed to potential pressures to have all computing done in one massive center, with a campus director reporting to a state computing czar rather than through campus channels. There is real potential for a problem here. Perhaps some of you find yourselves in a position similar to mine with regard to equipment acquisition. In my home state of Washington, every computer-related acquisition - including keypunches, terminals, and acoustic couplers - must be approved by the State Data Processing Coordinator, for all state agencies, including colleges and universities. We are fortunate: our state coordinator is a very able, knowledgeable man who has a broad vision of the role of computing, and he has assembled a staff that is intelligent and capable. I can imagine a less happy situation in which a man in such a position might regard his job as one of holding the line or minimizing costs without regard to benefits. I believe the writing on the wall is clear - as Computing Center Directors, we must be willing to consider state networks with an open mind, but having reached a conclusion, be prepared to document it. And I believe there are excellent reasons for a director to report through existing campus channels, rather than to a state czar; we may even find allies in our own administrations on this issue.

The final speaker, Bruce Alcorn, shifted our attention to the national scene, through calling attention to the tables in his paper, "The Role of Computers in Small College Management." Bruce, I did have a question about Table 6 on page 12 of your paper: does the first half of the table apply to fiscal year 1966-67? (Bruce Alcorn replied that it did). This is a <u>very</u> significant table: it shows that expenditures for computing in higher education more than doubled in the three-year period from 1966-67 to 1969-70. And the footnote calls attention to a particularly dramatic increase: for institutions with enrollments less than 2500, and offering no degree higher than the Bachelor's - i.e., that class of institutions that forms the primary representation here - the expenditures increased from $13.5 million to $44.7 million, a more than three-fold increase! But it is the total figure for all of higher education that is most likely to capture attention: $472 million in 1969-70.

The latter figure reminds me of the Pierce Report, and suggests a hazard that, collectively, we face. The so-called Pierce Report was published in early 1967 as a report by a panel assembled by the President's Science Advisory Committee, and chaired by Dr. John Pierce of Bell Labs. It considered the role of computing in undergraduate education, and presented a number of recommendations. The one recommendation that really caught attention was the one that, if memory serves correctly, expenditures for undergraduate educational computer use should reach a level of about $400 million in 1972, and that the federal government should provide about 3/4 of the total. Even in Washington, D.C., figures in the range of $300-$400 million are noticed, and I'm afraid this looked like the initial excavation on a bottomless pit - it scared the pants off some people, and led to a classical gambit: conduct a low-cost experiment to determine the feasibility of the Report's recommendation. The results are clear - the experiment is withering away and the recommendation is gathering dust. I regard this as a classical example of a revolutionary proposal that makes sense, but for which there was no adequate educational campaign in the right circles. This may suggest that we're playing with fire, but I wonder where we'd be today if one of our early ancestors hadn't been intrigued by flames.

SECTION II: PROGRAMMING SUPPORT AND FACULTY DEVELOPMENT

Chairman: Harris Burns, Jr., Randolph-Macon College

FACULTY DEVELOPMENT

Louis Parker, North Carolina Educational Computing Service

THE TRAINING OF TEACHERS IN THE USE OF COMPUTERS IN THE CLASSROOM

Pamela McGinley, Technica Education Corporation

PROGRAMMING SUPPORT AND FACULTY DEVELOPMENT: A VIEW FROM AN INSTRUCTIONAL COMPUTING CONSULTING SERVICE

Karl Zinn, University of Michigan

FACULTY DEVELOPMENT

Louis Parker
North Carolina Educational Computer Service

My organization, the North Carolina Educational Computing Service (NCECS), is a regional computer center currently serving forty institutions of higher education in North Carolina. We have been heavily involved in faculty training since our network began in 1967. After about two years of network operation, we realized that computer usage in disciplines other than computer science was very limited. We saw a need for efforts by the central staff to develop computer use in the physical sciences, the social sciences, business administration, etc. I would like to spend my time today not on technical training but on NCECS activities to promote computer use in a variety of disciplines -- an area we call "curriculum development."

We received a grant from the National Science Foundation 2-1/2 years ago to support a statewide program of cooperative curriculum development. The program has included collection and development of curriculum materials, introducing the materials to faculties and students across the state through workshops, documentation, and supporting the materials by both technical and information services. One of the first things we did was to create a position of Curriculum Development Manager on the central staff, and to obtain someone who was primarily an educator (with broad interests in computer applications) rather than a computer scientist. The program described below has been managed by Dr. Joe Denk, Curriculum Development Manager for NCECS, and the following description was derived from his reports.

Since 1 November 1969, approximately 700 different packages of computer-based curriculum materials have been collected from outside sources or developed by the ECS staff. Although many of the packages have potential uses in other academic disciplines, the materials have been categorized as follows: acounting, biology, business administration, chemistry, economics, humanities, interdisciplinary, mathematics, physics, sociology, statistics, and technology-engineering. A limited amount of material has also been collected in data processing and in psychology.

Collection of a package does not necessarily lead to full availability to a user. Those involved in transporting materials realize the problems created by differences in machines and programming languages, as well as the barriers to importing educational philosophies associated with materials. Therefore, NCECS may give a package either full or low support.

A package which reaches full support is available to run at the Triangle Universities Computation Center (TUCC) from a documented library. Low support means that NCECS has the materials but that they are not immediately available on the computer for one or more of the following reasons:

(1) A program may not be translated into a language supported by the IBM System at TUCC.

(2) Materials may not be of interest to more than a few users.

(3) Documentation is insufficient.

(4) Programs or materials are not completely debugged.

NCECS publishes a catalog-journal for each of the disciplines mentioned above. Program and Literature Service (PALS) was published for the fifth time December 15, 1970. That issue listed 392 packages available at NCECS, the locations of several packages not yet obtained, and a limited number of reviews of literature on computer-based materials.

PALS is published not only to inform users of the availability of packages, but also to inspire work toward importing packages into the network and to keep a running tally of the evolution of packages toward full support. It has produced requests from outside North Carolina for 1,000 programs and from inside the state for 300 programs. The August, 1971, issue of PALS contains about 700 packages.

Availability of materials is largely due to efforts of the curriculum development staff of NCECS. However, a significant proportion has been made available by faculty members and students under two funded programs of NCECS: a summer program involving full-time work (12 have been funded), and a small project program not involving release time (20 have been funded). These programs were geared to transporting and bringing to full support existing materials, as well as to creating new materials -- 76 of the 700 packages were created in North Carolina. The curriculum development staff also created a few new packages.

Priority for achieving full support was given to materials which run better (or could only run) on a large computer like that used by NCECS at TUCC. This priority did not exclude materials which also run on smaller installations, but it did reflect an emphasis on the unique educational aspects of a large computer. Among these aspects is the ability to store and handle large complex simulations, data bases, data handling systems, complicated tutorials, and statistical packages not available to small machines. Development was not limited to these large, complex systems, since a wide spectum of materials is necessary for education. Re-creation of this spectrum is impossible for each installation. Priority in support, therefore, was also given to materials thought necessary to produce this wide spectrum.

During the academic years 1969 to 1971, NCECS ran 32 workshops involving all of the disciplines in which materials are supported and several additional disciplines. These workshops provided one-on-one training in the use of over 200 computer-based packages to a total of 1173 participants. This meant two days of training in the actual running of packages for approximately 400 teachers and 100 students. Of the 104 institutions of higher learning in North Carolina, 76 were represented.

In the majority of the workshops, "group leaders" (teachers receiving summer appointment by NCECS) coordinated the activities of their own disciplines. When it was possible, authors of packages or experts in the use of the packages were brought in from inside and outside North Carolina to aid in the training. In this manner, not only were materials transported but the educational philosophies and theoretical bases of the packages were brought in also.

Since some of the participants had no previous programming experience, during most of the workshops the NCECS technical staff (assisted by some experienced users) gave short courses in beginning skills in programming languages, Job Control Language, and in the use of several packages. They also provided one-on-one training, assisting the group leaders. In this way, they provided user service oriented toward the discipline rather than merely toward computer science.

NCECS publishes two levels of documentation -- a brief form referenced as "LIB-ECS" programs and a teaching unit called a Computer-Based Educational Guide (CEG). The purpose of the LIB-ECS documentation is to give brief instructions for using a package and to touch on the theory and educational approach involved. A CEG is intended to amplify these instructions and the educational philosophy of the package. To data 33 LIB-ECS documents and 11 CEG's have been published.

NCECS takes a conservative approach to documentation; no documentation is produced unless a package has reached some degree of acceptability by teachers in workshop and classroom tests. This approach explains for the most part the availability of relatively few documents -- only 44 out of a possible 700. Almost 200 programs have been tested in workshops and over 100 have been used in the classroom. A total of 77 programs are actually available on the computer.

Another cause of the low number of available documents is the difficulty of getting documentation out of anyone. Almost 50 documents are now in preparation.

NCECS maintains an automated memoranda index which gives a documentation bibliography followed by a Key-Word-In-Context index. Another automated system maintains a user list by discipline involvement. The communications system provides easy dissemination of appropriate information and results in quick feedback from users. Currently, there is a lag of less than two weeks between announcements of and subsequent requests for available documentation. Workshops and meetings can be arranged assuming the same quick response.

This information service has taught NCECS a valuable lesson. Once direct contact can be made with active or potentially active faculty users, interested personnel become involved. Indirect contact with administrators and departmental chairmen did not produce much involvement during the early curriculum development activities of NCECS.

I realize that I have been talking about a large central facility having substantial NSF support (although the activities I described were designed to meet the needs of small colleges). The regional network offers one means of providing such services to the smaller institution. If there is general agreement on the need for this type of support on the small college campus, then we should try to adapt these concepts to the environment of the college with its own stand-alone equipment and limited resources. I propose this as a topic for discussion later in the session.

THE TRAINING OF TEACHERS IN THE USE OF COMPUTERS IN THE CLASSROOM

Pamela McGinley
Technica Education Corporation

Since the use of computers is increasingly common in school administrative and instructional programs, teacher and administrator training is of utmost importance. Teachers and administrators are being called upon, increasingly, to make important and costly decisions regarding curriculum content, computer languages, and equipment, as well as the administrative side regarding scheduling, grade reporting, budgeting, etc. Too often these teachers and administrators have had little or no appropriate preparation for this responsibility.

Traditional college and university courses in computer programming or data processing have not been suitable for the special needs of teachers or administrators. The typical course is FORTRAN programming with no exploration into the why or the how of computers, no discussion of appropriate teaching methods and materials, and no motivation in using the computer to extend instruction.

Part of the reason for lack of college level courses for teachers and administrators is that there has been in the past very little demand for such courses. Now, pressure is being applied to teacher preparatory institutions from two directions. Students entering college have had an introduction to computers in high school and are pressing for further education, and high school teachers are demanding inservice classes.[1]

There must be instruction to equip all teachers and administrators with the knowledge and materials they need. There are two distinct areas of training: inservice training and initial training in the universities and colleges. The experienced teacher is confronted with new topics, and university education methods courses need new elements.

In either initial training or training of established teachers, it would appear that the mathematics staff is best equipped to deal with computer related materials. However, it is not essential, and indeed it is undesirable, that a general course on applications be the sole domain of math departments. A general course demonstrates the broad relevance of computing to diverse human activities, and hence involves the social impact of the computer. With regard to the training of established teachers or administrators, the approach to the material is different than in university courses.

It has been suggested that the introduction of computing to teachers might be patterned after the same methods which they themselves use in their classroom. It is most important, however, to deal with the subject in more detail than the teachers might do with their pupils. These teachers have an intensive knowledge of the school situation and are generally much more critical of the innovation of computing.

To run a course aimed at being an education course, rather than a computer programming course, can be a harrowing experience; properly run, it need not be. Once the teachers have grasped the basic ideas of computing, critical and most worthwhile discussions arise as to what we are trying to do and why. Personally, I find these discussions in this area most stimulating and consider it quite essential for anyone in the area of curriculum enrichment.

It is very easy to complicate the teaching of computing. However, only logical thinking is required. Important recent developments in interactive terminals, mark sense card readers, and conversational languages have made computers easy to use in the school environment.

Conversational languages, most notably BASIC, have made the biggest impact on our approach to computing in schools, and therefore to teacher training. It is now no longer necessary to teach a complex high level language which will take some days to master. A conversational language can be used interactively after an hour or two of study. This fact has removed the one major educational objection to computing in schools, thereby allowing the computer to be used as a tool in enriching curriculum.

Over the past few years the United States government has funded a number of research studies into the use of computers in education. Unfortunately, very few of the results have really been readily available to the administrator and the teacher. An exception to this situation is Program REACT - Relevant Educational Applications for Computer Technology.

REACT includes a series of courses developed by the Northwest Regional Educational Laboratory under the direction of the Office of Education. Several years were devoted to development and field testing of three training courses for teachers and administrators. These computer-related education courses emphasize educational computer applications and provide "hands-on" use of computers.

The series of courses demonstrates ways the computer can be used in school instruction and administration. As a result, school personnel increase their understanding of the problems and potentials of using computers. Intelligent selection can then be made from the growing number of possible uses of the computer in education. These training courses are organized in separate modules for school administrators and teachers.

REACT's Course I - (Administrators and Teachers) - COMPUTERS IN EDUCATION: A SURVEY. This course provides administrators, teachers, curriculum specialists and undergraduate and graduate educational majors an introductory familiarization with computers. Through the study of elementary computer concepts and the role of the computer, the educational uses of computers are presented in a broad context. The teachers and administrators develop an understanding of:

Concepts of computer components, input-output, storage and differences in computer types, generations, sizes, and speeds.

How man communicates his problem to the computer for solutions through different types and levels of programming.

The concepts of mini-computers and time-sharing, the use of the teletypes for on-line introduction and elementary BASIC programming.

Teachers and administrators are introduced to the fields of educational computing and prepared for intensive study of classroom uses or administrative use of the computer through presentation of the over-view of the field. Understanding is developed for the potential of computer use in classroom problem solving, vocational training, computer-assisted instruction, simulation, library management, guidance and training, curriculum management and integrated data management systems. There is classroom presentation and "hands-on" experience with individualized instruction.

REACT's Course II (Teachers) - COMPUTER-ORIENTED CURRICULUM. The course offers a complete description of application units available in Social Studies, English, Business, Sciences and Mathematics. This includes a comprehensive review of the resource materials currently available. Materials augment present course outlines; that is, the program supplements and enriches existing learning and teaching methods -- it requires no change in curriculum. Curriculum is extended because of increased efficiency and capabilities inherent in computer use. Each application unit from the various subject areas includes:

A description of the computer program.

A rationale for the unit.

Suggestions for several ways the unit could be integrated with the on-going curriculum.

Objectives describing the desired student performance after each unit.

Required preparation for students planning to use the unit.

Directions for further study and exploration of the concept.

A complete computer program run.

Emphasis will also be given to student-oriented materials, laboratory and demonstration techniques and inter-disciplinary approaches. In addition, the course will include a continuation of the elementary BASIC instruction offered in Course I. Instruction will be completely individualized and tailored to meet the specific requirements of each participating teacher or staff member.

REACT's Course II (Administrators) - COMPUTER APPLICATIONS/ADMINISTRATORS. The course utilizes a demonstration data management system in a "hands-on" environment to explore a variety of fundamental administrative data processing concepts. A system is constructed for a model school in order to examine major application areas. The applications range from the routine (preparation of report cards) to the imaginitive (a program planning budgeting system). General topics of study include:

Recognition of an effective computer application.

Traditional educational administrative applications as implemented on a computer.

The computer as a decision-making and planning tool for school administrators.

Basic computer functions as applied to educational applications.

Opportunities and problems presented by a computer.

State of the art.

The administrator will have a broad picture of the types of tasks within a school district that are best suited to a computer and an appreciation for what is involved in implementation. Implications from a management standpoint will be discussed. Specific areas of interest of individual participants will be addressed as time allows. Classroom and workshop environment.

Each of the classes just described utilizes a series of publications developed under the REACT program. The manuals are also available individually or in course sets for use as texts in university education curriculum, or for individuals who wish to use them in a self-study setting.

Information about the REACT program is available from:

Duane Richardson, Director
REACT Program
Northwest Regional Educational Laboratory
500 Lindsay Building
710 S.W. Second Avenue
Portland, Oregon 97204

Technica Education Corporation
655 Sky Way
San Carlos, California 94071

Initiative and enthusiasm are the only prerequisites for being involved in computers in education. Obviously, I think that all teachers and administrators must be involved with the computer as soto as possible. This imposes a tremendous challenge on our educational training system and it is one which we must tackle with all resources.

[1]Holznagel, D. C. and Stonebrink, G. "Inservice Education, The Oregon Computer Instruction Network Experience.", Data Processing for Education, 1970, Vol. 9 No. 7, 1-2.

PROGRAMMING SUPPORT AND FACULTY DEVELOPMENT: A VIEW FROM AN INSTRUCTIONAL COMPUTING CONSULTING SERVICE

Karl L. Zinn
University of Michigan

Three purposes have motivated this presentation. First, I would like to provide information about two projects in which I am involved in Michigan. The one of most interest for this symposium is Project EXTEND, a small college consulting service for instructional computing. The second is the MERIT network computing project, a prototype for a regional computing service in Michigan. Further details about both of these can be obtained from the references (Zinn et. al., 1971; Herzog, 1972; Carroll et. al., 1971).

Second, I would like to report some of my experiences with user services and faculty consultation. Although my experience has been primarily with University of Michigan faculty, recently I have extended these services to the small college environment.

Third, I shall offer my views on some contributions of engineering, informational and behavioral sciences to computers in college teaching and learning activities. My prejudices as a behavioral scientist should be clear; perhaps it is useful to know that I was trained as a psychologist, I teach computer science, and I do most of my consultation and service work in education.

To generate interest in faculty consultation in a few short pages I shall be quite explicit and perhaps controversial. For supporting arguments, the reader might consult the publications from which this material has been drawn (Zinn, 1970, 1971, 1972). I hope I will hear from those who would exchange ideas and materials with either Project EXTEND or MERIT.

SOME PREFERENCES AND BIASES

I shall take up my third point first in order to expose some of my biases about what a computing center and its related offices should provide to college faculty. Most important, the computing capabilities should be very accessible. Simply to provide interactive computing is not enough. A number of university computing centers acquired NSF funds to establish "a regional computing network." Initially, these were not networks, except in a degenerate sense; they did not offer service! The funds were used by a computing installation to improve a facility; only incidentally were partial services offered to nearby colleges.

Simply to make computing interactive and somewhat available through telephone lines and remote terminal devices doesn't make it useful for instruction. Furthermore, a university computing center is continually changing its system and software; much of the computing software is not particularly suitable for instructional uses; and documentation is quite sparse and usually ineffective. Consultation at a remote site has been almost non-existant except for two or three isolated successes of regional services. The current funding of regional service activity seems to be correcting many of these mistakes.

Acquiring interactive computing through a commercial service does not offer much better chances of instructional use. The software may be more stable, and some of the documentation somewhat better than in university settings, but the capabilities are not particularly suitable for instruction in most cases. In-house, multiple-terminal systems are not clearly the answer either, at least at the present state of software development.

Although it helps to make computing more accessible by providing good software and manuals, I must add that to provide a simple programming language is not enough. Dartmouth College has assembled impressive statistics on computer use, but conversations with faculty there and reviews by professional colleagues elsewhere raise questions about the extent of contributions to instructional computing. Later versions of BASIC and special packages such as IMPRESS (Meyers, 1969; Denk, 1972) have made more of a contribution to computers in education. However, the mere existence of facility for simple programming has not led to intelligent use by any efficient route. Neither have manuals written for specialist users of package programs provided sufficient guidance for novice computer users.

I visualize the siutation with the needed computing capability at the center of successive levels of user support. Software and manuals make up the first level of effort to reach the user. A second level I call "user interface considerations". Significant ideas can be drawn from engineering, information and behavioral sciences to build effective tools for instructional computing. The expertise of a number of different scientists is involved in the intelligent selection or design of convenient terminals, command language, file structures and other aspects of the interface with the user.

A third level provides even more convenient access for the user through applications programs, packages and convenient user documentation. A fourth level in my view involves (Figure 1) general education about computers and the training of specific skills for users. These last two areas, applications and training, will serve the majority of users on a small college campus; a small minority are interested in programming languages and technical manuals.

By the set of concentric circles in Figure 1, I mean to represent a perspective which might help some computing center directors service users. Some other diagram might be drawn to give the view of a faculty member or applications programmer. I want to point out here the additional services and considerations which must be added to computer availability to make it convenient and appropriate for instruction. Incidentally, I am not a proponent of interactive computing for instruction regardless of the application; in a number of situations I think it is much better to punch cards or enter commands and leave the system to do its work and return results in some convenient (probably batch) mode of output. In face, the distinction between interactive and batch will decrease in the coming years until it no longer is a relevant dimension. Users will care only about distinctions between interim and final results or transitory and permanent records.

For this audience I want to emphasize that the behavioral scientist makes an important contribution to instructional computing in the careful analysis and evaluation of instructional needs and computing resources. A pattern of resource development has evolved at the University of Michigan in which the student is introduced to computing through a simple programming application which typically is presented as a gaming or management situation. Before he tries of this game, the computer program or the teacher encourages him to look beyond the game at the underlying model or simulation on procedure. Most students then become interested in learning more about uses of computers in their discipline; some seek additional training to acquire specific skills and information for other uses of computing. The arrangement of resources is diagrammed in Figure 2. A similar succession of experiences and another diagram of resources has been derived from experience with the teacher-author.

My differences with some others working on instructional computing in college teaching may come down to a matter of strategy more than educational philosophy. Some say it is important to get each student to program at a very basic level, developing each new application according to his own conception of the procedure. Others say it is better to get the student to look at real examples of computing in the discipline being studied, attending to procedures at the user level rather than the programmer level. In fact, it should be possible to achieve some balance between these two views. My diagram of resources (Figure 1) should not imply any kind of rigid boundaries or comprehensive set of opportunities. One should encourage the faculty member or student through whatever kind of computer use gets him started, and then help him to move from that kind of computer use to another and from one resource to another as may suit his needs. A teacher may become interested in an expensive kind of desk calculator, but, because of a convenient user interface in the system, then try other package programs for social science data processing; such a computing center has done its users a good service. Furthermore, an effective arrangement of opportunities will help a faculty user to become acquainted with how computers are affecting the lives of his students in general, as well as how they influence research and scholarly work within his discipline.

PROJECT EXTEND AND THE MERIT NETWORK

The Michigan Education Research and Information Triad (MERIT) has established a computer network experiment. Information about this project is available in the progress reports and other publications listed in the bibliography. It has been working on engineering problems for some time, and so far its major contribution to small college computing in Michigan is one of communications and the assembly of resources.

If the MERIT network is successful, and so far the progress has been promising, small college users of computation and information files throughout the state will be able to draw on the greatly enriched pool of three or four large university computing centers. MERIT planners do not propose that all computational needs will be met by the network facility; nevertheless, certain important resources for instructional and research programs will not be economically achieved in any other way.

Project EXTEND was established within the MERIT environment to bring demonstrations and trial experience with computing resources to small college users. It draws from MERIT an inventory of rather unique computing and information resources, attempting to package them for convenient use.

Project EXTEND also draws upon the various computing centers, but particularly the University of Michigan Center, where the Michigan Terminal System (MTS) originated. MTS is a rather usable computing service; it provides both remote batch and terminal access for college users with a very rich library of software on an IBM 360 dual 67 (Boettner and Alexander, 1970). A second version of MTS is being used at Wayne State University and also includes administrative data processing.

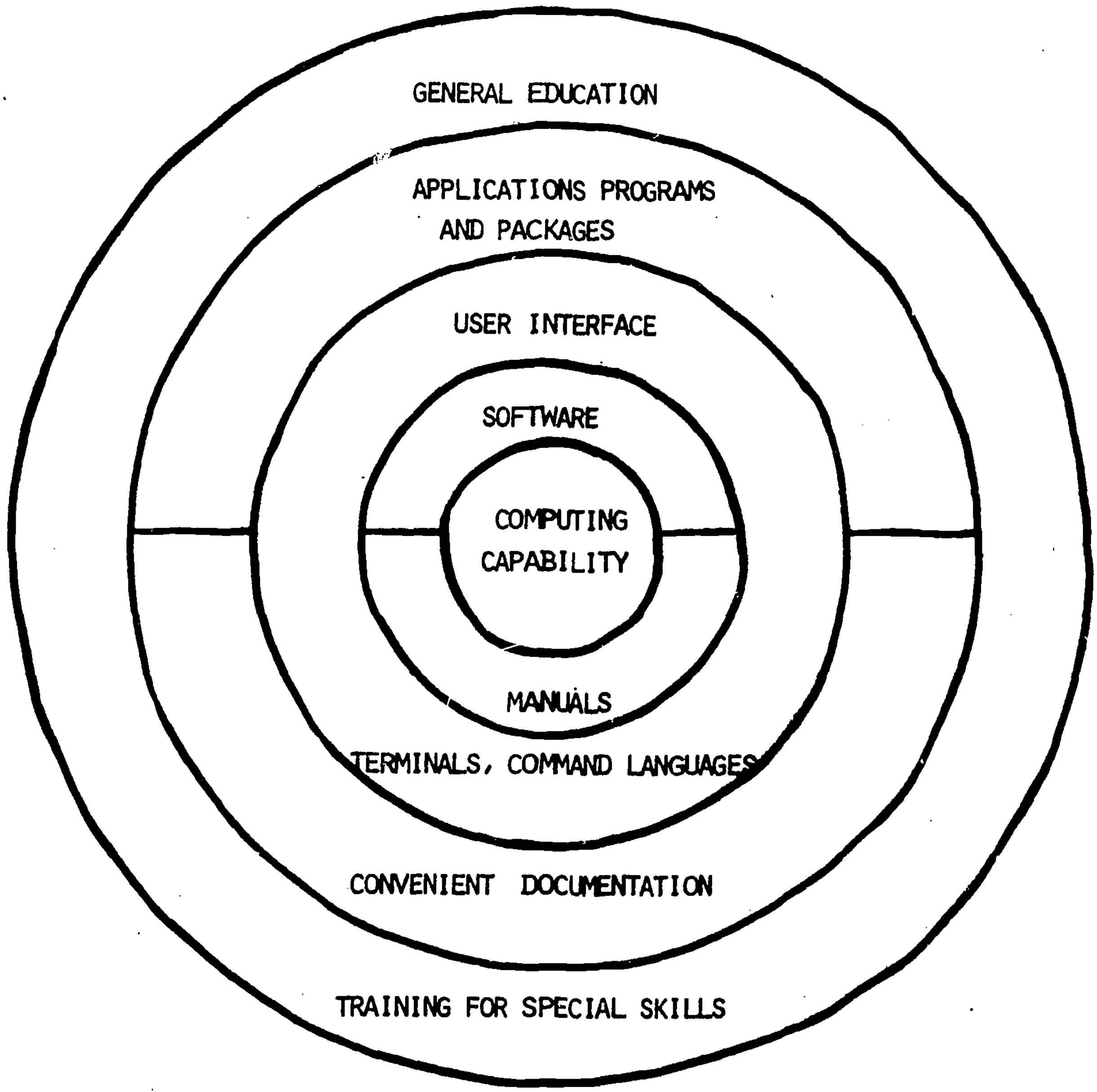


Figure 1: A Perspective on Service to Users

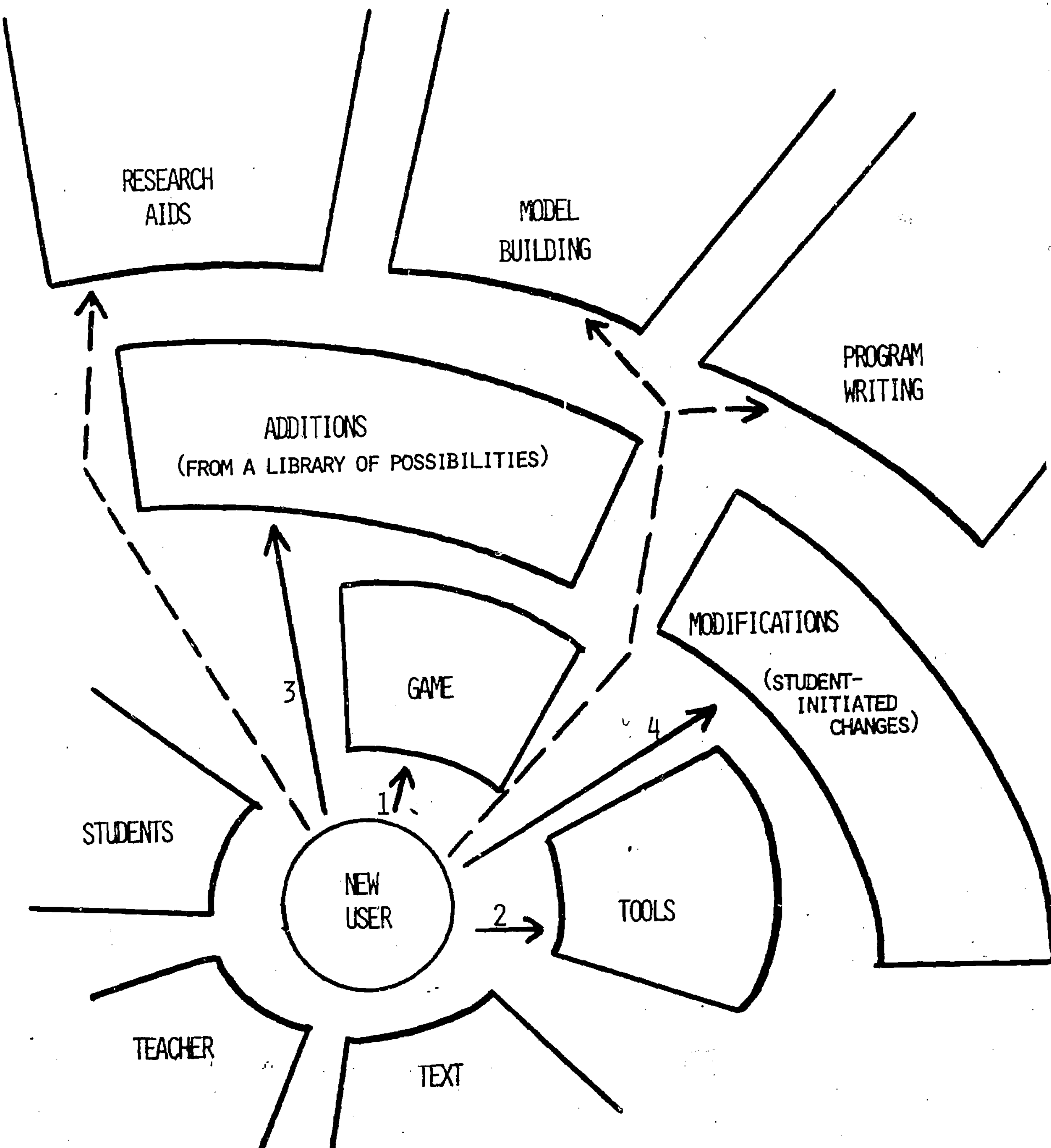


Figure 2: An Open-Ended Approach to Student Involvement with Computing Activities

Project EXTEND draws heavily on the software and documentation provided by these computing centers; however it has attempted to adapt some of these materials for small college users. For example, the documentation on MTS is given in ten volumes; an EXTEND staff member (Davis, 1971) reduced it to a 50-page primer. EXTEND staff are also working on task-oriented summaries of different program packages, including an abbreviated guide to use of the terminal, some parts of which can be cut out and stuck on the various terminal devices as reminders for proper use.

Project EXTEND also draws on the resources of the Center for Research on Learning and Teaching, which has been consulting for ten years on various matters of learning and teaching, including instructional use of computers. This office has been the primary source of demonstrations of instructional computing for small college uses. However, its materials must be rewritten and adapted to fit the remote computing environment and to be usable outside the University of Michigan.

The most important contributors to the success of instructional computing are the departments. The ideas and judgments of people in the various disciplines are crucial to acceptance of new technology for learning. Project EXTEND staff have relied heavily upon teachers in the various disciplines. Not only is their judgment on the value of various applications crucial, but their advice on documentation, user guides, and other support materials is quite important.

I feel very strongly that the contribution of the disciplines cannot be handled in some other way, for example, through computing centers, the administration, consortia, publications on instructional computing or whatever else does not include the leaders in the teaching of each discipline apart from computers. The decisions will be made, and the incentives will be provided, by those people who are leaders and set quality standards for what is important in the discipline and its teaching. Perhaps for a time, the field of instructional computing will be aided by a journal on techniques for undergraduate education; perhaps it would be handled more appropriately by an annual conference, a proceedings and similar one-time publications. However, the publications activity must be picked up by the teaching journals in various disciplines if it is to be recognized and survive.

Initially, Project EXTEND was named to suggest the extensions from research to instructional use of computers, and from large university centers to small colleges. For instructional computing, at least at the large university, a relation between research and teaching uses is very valuable: use in teaching is easier and more likely for a professor who makes similar research use of the computer; some research uses benefit through new ideas and programs developed for the instructional use which can be applied back in the research activity. Almost before Project EXTEND began, it became clear that it might as well be called Project EXCHANGE. Current computing activities at small colleges include as significant instructional uses as at the large universities. The project will make a major effort to pass around ideas and programs from one institution to another, considering the small colleges to be among the most important sources.

The Project has nothing to sell; it offered time on the Michigan Terminal System only because MTS was convenient and available for demonstration with a variety of examples of instructional computing. I try to hide my preference for large systems when I go into a small college for consultation; if I have any bias it is against providing the major amount of instructional computing at a small college through a large university center. I would like to see a situation develop in which the large universities which support research provide the computing procedures and program packages, and the smaller universities and colleges which excel in teaching will provide the pedagogy.

SERVICES AND FACULTY CONSULTATION

Project EXTEND is attempting a translation of services at a large university to small institutions in the area. Small college computing centers must look for cooperative arrangements to assemble resources and expertise. I have argued that many different scientists have something to offer, and a small campus is not likely to have sufficient staff or the particular expertise required. Some of the services and support functions which Project EXTEND provides are listed below; the interested reader should look into our reports for details.

1. Information. The files and technical memos of the project provide information and advice about the capabilities of computers for use in instruction. The small-college administrator or professor is referred to various resources: first those which may already be available in his institution or from a service within the state, and then to external sources such as newsletters, national directories and professional meetings. We try to make information easy to access and rewarding to use.

2. Demonstrations. A number of rather general demonstrations have been prepared, each one indicating another kind of computer contribution to learning. We try to adapt these to the individual if he is really serious about examining computer contributions to his particular subject and his way of teaching. That is to say, we have consulting staff and a programmer who will adapt and extend a particular demonstration along the lines suggested by a potential user. Sometimes this leads to regular instructional use of the demonstration as modified by the professor. Again, we try to make access to these demonstrations easy and their use rewarding.

3. Consultation and Training. Project EXTEND tries to provide advice and training in the context of the goals and needs of the students and their institution. Nearly every activity of the project involves improving the learning resources for individual students. We advise individual faculty and provide training through workshops and written materials, emphasizing means for carrying on effective instruction. The development of computer-related instruction materials is a primary subject of workshops and consultation; advice is offered on development procedures, personnel requirements, appropriate equipment, evaluation of outcomes, funding of further development activities, etc. In some cases, we give specific training in programming or instructional design procedures. Project EXTEND provides consultation and training at little or no cost to the institution or the individual, and the major cost to the participating college is personnel time. We do encourage institutions to provide, as an incentive, time release from other duties; hopefully the support of one's peers also provides an incentive and is forthcoming if the work is of quality.

4. Programming Assistance. Project EXTEND offers programming to meet the specific and individual needs of participating faculty and institutions. The staff work is done through a notation for describing instructional procedures which is somewhat independent of any programming language. This becomes the documentation for the instructional procedure, and a medium for communication between programmer and teacher. To be successful in this area, we try to be very responsive to the ideas put forth by the faculty member. He is encouraged to make suggestions, observe their implementation on the computer and test them out with students. If it is easy for the designer of exercises to see a change followed through to execution, he seems more likely to make program changes as the need arises.

5. Reproduction and Distribution. Project EXTEND provides special equipment and services that may not be available on individual campuses. Faculty may generate animations by computer and find the production of short films an effective way to convey graphic representations to students in the classroom or individual study facilities. Supplementary written materials for computing exercises can be duplicated with Project EXTEND facilities. Perhaps more important, the staff of Project EXTEND assist with editing and production of manuals. It does seem that the quality appearance of documentation is a great incentive to do further and careful work; furthermore, a specialties office such as that associated with Project EXTEND can do the work more economically than each individual institution.

6. Evaluation and Reporting. The project provides information about local and national publications which might carry reports of instructional computing at participating schools. We encourage careful attention to evaluation of the activities, including objective measures of performance or reports of student attitude wherever possible. Perhaps more important is the professional review by peers in the same teaching area. The incentives for reporting activity are academic credits for publications and the increased likelihood of promotions or salary increases.

These ideas about services and support are still evolving through Project EXTEND activities. For current details and examples, the interested reader should write to the Project EXTEND office.

SUMMARY

I have suggested that a consulting service at the small college or within some regional consortium has to take what the computing center or other computing services have to offer, consider the needs and requirements of the people in various departments, draw out the advice and assistance of the instructional resources facility, and bring these resources together for the purpose of more effective instructional computing. This combination has been accomplished to some extent by the staff of Project EXTEND (Figure 3).

For example, we rewrote the system documentation because we did not want the participating teacher to have to spend a week or a month learning about the system. We still have to provide useful updates to the participating teacher so he will not waste time maintaining current knowledge of the system. We write brief guides to use of the terminal, with stick-on reminders at various places on the keyboard so that the student, who may have only brief experience with the computer during a course, will not spend more than a few minutes learning the typing procedures and the commands of the system.

I have pointed out the importance of determining whether the computing activity is really contributing in a substantial and economic way to the learning and teaching resources of the institution. Many decisions about the resources at an institution are made on an ad hoc basis, such as deciding whether a library should buy a certain set of reference works. Increasingly, decisions are made with a more careful analysis of the needs of learners and the specific contribution of the resources under consideration. Because computing is believed to be expensive, and perhaps a luxury, it comes under more critical examination. Procedures and tools can be borrowed from other areas of science to assess the contribution of the computer. One of the products of Project EXTEND will be a set of guidelines and checklists for deciding where and in what way to use the computer, and how much effort to spend on developing particular applications.

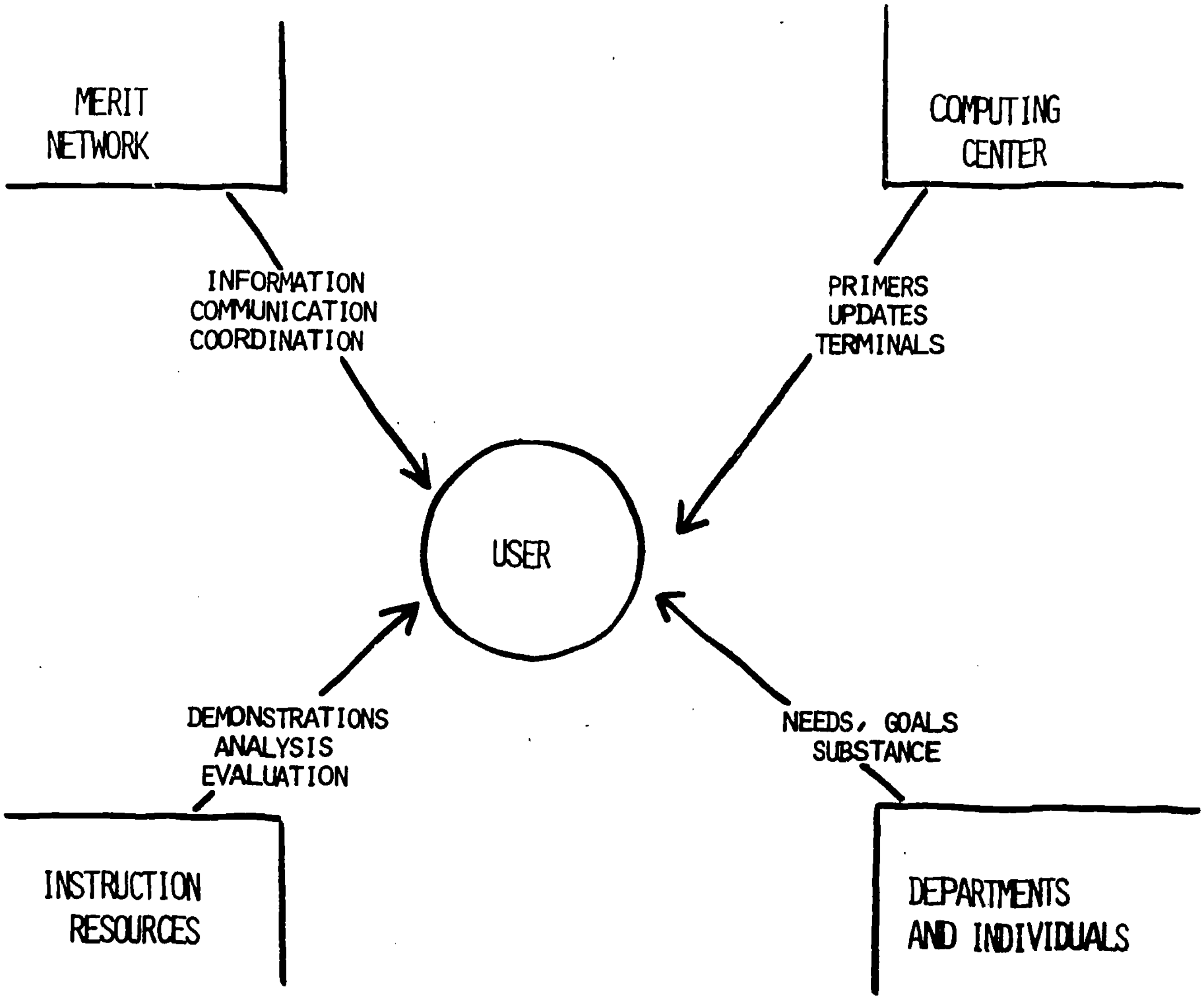


Figure 3: Combining Resources Through Project EXTEND for Instructional Computing

I do not have answers to the important questions such as whether a small institution must put resources into an instructional service center, or whether some consortium arrangement can provide the same or better service for less money. I am sure, however, that some kind of validation procedure and some kind of professional review of the proposed computing activities is needed if we are to be sure to move forward in the field of instructional computing.

BIBLIOGRAPHY

Aupperle, Eric and Becher, William D. "The Communications Computer Hardware of the MERIT Computer Network," IEEE Transactions on Communications Technology (in press).

Boettner, Donald W. and Alexander, Michael T. "MTS - Michigan Terminal System," Operating Systems Review, pp. 6-19, ACM SIGOPS, Vol. 4, #4, December 1970.

Carroll, Tom W. and Wolfson, Seymour J., and Zinn, Karl L. "Progress on Applications Development, 1970-71: A Report of the Associate Directors of MERIT," MERIT Computer Network, Ann Arbor, December 1971.

Davis, Wayne K. (Ed)., A Short Guide to MTS and Remote Terminal Operation, Project EXTEND, Center for Research on Learning and Teaching, University of Michigan, Ann Arbor, 1971.

Eick, Harry A., Wolfson, Seymour J., and Zinn, Karl L. "Development of Applications for the MERIT Computing Network," Bulletin on Computer Uses in Education. pp. 7-11, ACM SIGCUE, Vol. 6, #3, June 1972.

Evans, Barton A. and Lipschutz, Marlene E. (Eds)., Demonstrations of Computer-Based Learning Aids. Project EXTEND, Center for Research on Learning and Teaching, University of Michigan, Ann Arbor, 1972.

Herzog, Bertram. "Computer Networks," Proceedings of the International Computing Symposium, ACM and AICA, 1972.

Zinn, Karl L. "Computer Learning Under Evaluation (Project CLUE) and an Attempt at Hyperspeech." pp. 135-144 in Computers in Instruction: Their Future for Higher Education, Proceedings of a conference held October 1970. Rand Corporation, Santa Monica, California.

Zinn, Karl L. "A Technology for Instruction in the Age of Informatics," in Fortschritte and Ergenbnisse der Unterrichtstechnologie. Report of the Congress "Lehrsysteme 72" held in Berlin, April, 1972. Munchen: Enrenwirth-Verlag, in press.

Zinn, Karl L. "A Personal View of Computing Software for Use in the Instructional Process." pp. 402-412 in Proceedings of the International Computing Symposium, ACM and AICA, 1972.

Zinn, Karl L. et. al. Project EXTEND: An Introduction. Center for Research on Learning and Teaching, University of Michigan, Ann Arbor, 1971.

SECTION III: ADMINISTRATIVE/ACADEMIC INTERFACE

Chairman: R. Waldo Roth, Taylor University

COMMENTS ON THE ADMINISTRATIVE/ACADEMIC INTERFACE

Jacques La France, Wheaton College

COMMENTS ON THE ADMINISTRATIVE/ACADEMIC INTERFACE

Ronald Anton, Swarthmore College

COMMENTS ON THE ADMINISTRATIVE/ACADEMIC INTERFACE

Jacques La France
Wheaton College

Wheaton College began with data processing twelve to fifteen years ago with the use of a 402 accounting machine for the college accounting system. This was soon replaced with the 403 and then a 407 as the applications began to expand beyond the business office into the area of student records. In time the 407 was combined with a 1620 computer, and with the acquisition of the 1620 computer students soon became involved.

For a few years, the administration and the academic areas used the 1620 jointly, with a scheduling of each group to particular time periods. However, by 1969 the use of the 1620 by both groups had become so heavy that at certain time periods there weren't enough hours to go around, and each group tended to need to have the computer when the other one was supposed to have it. Students would need it to get class assignments done and the administration needed it to get out some reports. At this time, the college decided that the only effective solution was to get two separate computer systems for the college, one for each group. Hence the 1620 and 407 combination was replaced by a 360/20 for the data processing department and an IBM 1130 was ordered for the academic departments to use. I do not know what the organizational plans were for the administering of the 1130 operation.

Prior to the installation of the 1130 the college was given a PDP-12 computer on the condition that they would hire somebody on the staff with training in the area of computer science to give the college leadership in the use of the digital computer in academic disciplines. The college accepted this and at that point the Academic Computer Center was established, and the director of this new center was hired in the spring of 1970. With the establishment of the Academic Computer Center, there were then two separate computer centers on the campus, separately administered and reporting to separate administrative officers. The department of data processing, which handled all the administrative use of computers, reported to the college treasurer through the business manager. The academic computer center handled all the computing needs of the students and faculty and reported to the Dean of Faculty.

Since I'm the director of the academic computer center, I've only had experience on the campus under this two-department system. However, I find that this has worked very well in that I have been able to devote my full attention to the academic computing needs and have not had to concern myself with the administrative computer needs and the maintaining of the college records. Also, the students have had essentially unlimited use of the computer facilities in the academic computer center because there has been no need to work around the schedule of the administrative needs. As a result, all the computing is done hands-on, primarily through running a small time-sharing system with 7 terminals for about 4 to 8 hours each day with the rest of the time period in each day available for single-person use of the computer.

There are some changes for the future that will affect this two-part system, however. The college has outgrown the capacity of its 360/20 with only 4K of memory, since it was still being used with the same operating philosophy that was developed for the 407 accounting machine. The decision was made to replace the 360/20 operation with a PDP-11 time-sharing system, with a completely new philosophy of data processing. In this new operation the various administrative offices will have terminals into the PDP-11 system, and data entry will be by way of these terminals from the department that has the data to be entered, thereby bypassing several intermediate steps of transmitting the information to someone in the data processing department for keypunching. Besides cutting out several intermediate steps it is expected this system will also improve the reliability of the data, by placing the responsibility for the accuracy of the records with those that are responsible for gathering and using the information. Third, it should increase the availability of the information since through the terminal the interested department can access the data on a moment's notice to retrieve whatever information is needed from the data bank.

It is also expected that terminals on the PDP-11 will be made available to students. This will help absorb the increasing student computing needs which are nearly to the point of exceeding the capacity of the PDP-12 to handle them. This means that we are moving back toward a system which is used by both administration and academic departments. However, in this case, because it is a time-sharing system, neither group should interfere with the other as long as they have separate terminals or there is an adequate number of terminals. There is no firm plan at this point how the administration of the PDP-11 system will be maintained, other than the fact that the manager of data processing and I have an extremely good relationship.

In summary, I believe that it's better for the academic program to have two separate computer systems, or have one system which is a time-sharing system so that the students have essentially unlimited use of the computer, limited only by the demand of the other students, enabling them to make use of the computer system at any time. Having a single batch system, I think, is not as desirable, but in many cases may be necessary simply from the economics of the situation, in which case it would be better to have a single batch system than no system at all. But if possible, I would recommend having a separate system for the academic use or having a time-sharing system.

COMMENTS ON THE ADMINISTRATIVE/ACADEMIC INTERFACE

Ronald Anton
Swarthmore College

Swarthmore College is a small liberal-arts college, located in southeast Pennsylvania. The attitude of the administration towards the computer until just recently was strictly hands-off. If you walked into an office and said, "How about putting something on the computer?", they would get up and walk out, or they didn't hear so well that day, or something like that. At present, we are doing a lot of administrative work on the computer, mainly because the vice-president put out a letter that said, "Use the computer." That was all there was to it.

We are starting to bring in the general ledger and accounting system for the business office, the alumni mailing list, which is about 22,000 names, and probably next month we'll do our first mailing, and numerous other small jobs which fell our lot because Miss SO-AND-SO who's been with the college for fifty years finally retired and nobody knows how to do the work she was doing. We print things like payroll labels, which they used to print by hand once a week on the time cards--about a three-hour job that we do now in about thirty seconds. Other things are vacation and sick leave, which is a monstrous problem, mainly because nobody else can figure it out, or wants to be in charge of finding the sick leave every month. They blame it on the computer if it's wrong.

Primarily, the reasons that we switched a lot of the work to the computer were (1) the grant which had specified that the computer was to be used for academic purposes had expired, and (2) because of present economic conditions the administrators felt that we needed information up to the date, rather than have it two years later, when we finally got around to getting that "new" analysis, which was out of date long ago. So that's what we are now working toward, to create new data-base files to do these reports that previously had to be done by hand by somebody going through accounts and taking off numbers. The computing staff at the college now consists of four people. It was one for a long time. We have a Director of Computing Education and Activities. His primary purpose is to be a politician. I go around and ruffle the feathers, and he smoothes them down and takes them out to lunch and talks about their tennis game, etc., and gets them settled down again. There is the Director of the Computer Center, who primarily keeps the place running, ruffles feathers, writes programs, and handles the operation of the computer center. We have a secretary, who does more computer operation than secretarial work. She comes in very early in the morning and does all the administrative running. And there is a keypuncher, who is a keypuncher fifty percent of the time, and also goes around to each department, discusses with them the newer projects, how they should lay out their work, and sort of helps them over the rough spots, like "what is a card?" and things like that.

We operate the computer center from six a.m. to midnight daily. Six a.m. to noon is the administrative time, and it works very nicely, when somebody--not myself--comes in at six a.m. and does all the administrative work. Nice and quiet. The telephone doesn't ring at six a.m., and you can get a lot of work done. From noon to five we're open for students. It is necessary to set a time limit on the machine which will kick them off after any time interval that we set. Usually, if it's very crowded, we set it for five minutes; if not, we just leave it alone. From five to six is usually reserved for one-hour runs or faculty. The faculty are interested in APL, and they come in from five to six. Six to midnight is reserved again for students, and the center is operated by student supervisors. We usually have two a night, mainly because they can't hack six hours at a crack, so they switch off. This has run very well. Ever since the day the computer came in there have been student supervisors. Students are very good at programming and can answer almost any question. In fact, I sometimes even have to call them up and bail myself out. We've spent a lot of time improving the efficiency of the operation. We have about four-fifths of the operating system. When we started playing around with it, IBM came around with a little piece of paper that you have to sign saying you won't turn any APAR's in. We send them in anyway.

Also, we like to soup packages up. We are presently working on a re-write of RELCV. We've got it running 300 percent faster than when it originally came. We operate the FMU compiler, which has increased efficiency in operation and also is a great aid to students in writing programs with the logical statements.

I have a problem: How can the computer center explain the operation of the computer to administrative personnel, who (1) aren't interested in the computer and are being forced into using it, and (2) still think the old way is better? This is sort of like the problem yesterday with the faculty, but it's a little harder with the administration since they haven't been in a classroom for forty or fifty years and are not about to start. If there are any solutions, or if anybody has partially solved this problem, I'd be interested in hearing about it.

SECTION IV: COMPUTER CENTER POLICY

Chairman: Bruce Alcorn, National Laboratory for Higher Education

COMMENTS ON COMPUTER CENTER POLICY

Richard Vogel, Western Maryland College

COMMENTS ON COMPUTER CENTER POLICY

James McDonald, Morningside College

COMMENTS ON COMPUTER CENTER POLICY

Richard Vogel
Western Maryland College

It looks like I will be touching on some of the topics that have just been mentioned, because some of them will be affected by whatever policy people might use for obtaining or selling data-processing services outside of the college environment. First of all, I'm sure each of you at one time or another has considered this possibility. It looks immediately attractive if for no other reason than that it brings some extra income to the college, thereby defraying somewhat the cost of the hardware and staff that you have in your center. Another advantage that you can achieve by selling some time and getting a little extra income is that perhaps you could support a somewhat larger computer facility than you might otherwise be able to afford. Eight more K of core is awfully attractive, or an extra disk drive or something like that can be real, real handy. If you can drum up enough business to support this sort of thing, you can put yourself in real good shape.

The other area would be in personnel. With the extra income you can support additional personnel. I'm not going to identify the particular uses that we have made--unless I'm asked--in selling services, but I'm just going to give you the general implications of a decision to sell data-processing services outside the college.

First of all, please be aware that you are going into business. You are no longer operating in an academic environment. The reason I say that is that you will now be faced with the problems of the production, distribution, and marketing areas of business; and there are problems in each of these areas that you must consider, or you'll get yourselves into very hot water. First of all, in the production area, you have to provide some type of programs for your outside users to use. These can be in the form of packages which you can create within the computer center, which you will then sell individually to different concerns, or you may choose to go on a custom programming basis, where you will approach people and see what they need, and will then respond accordingly. In either case, you must be prepared to devote some of your resources to writing these programs, either the initial packages or the custom programs as they come along. This is an open-ended operation; once you have taken this route, you must accept the fact that there will be improvements that must be made to your programs. If it's a package, you'd like to enhance it; if it's a custom program, you know perfectly well that if you give the guy this column of figures, he wants three more next to it.

You will also be faced, in the maintenance area, with having to make changes in the programs that you have written for your outside users, changes that are not caused by the requirements of these users; namely, if you change your hardware--go to a larger system--change manufacturers--you must bring your customer's programs along with you. If you change your operating system for one reason or another, or the system approach that you have, you will have to include them when you consider the programming changes.

If you're going to sell computer time outside, your customers are going to want their payroll to be run on time, and it will be very difficult for you to convince them that there is a student who is working on a computer science project at the moment and the payroll will have to wait until tomorrow. You will be forced to do job scheduling if you have not done this already. Another feature of selling computer time, particularly because you are a college or university selling the time, is the concern of your customer with the security of his data and his program. He has been reading the newspapers the same as you and I, and is convinced that every college and university is a hotbed of potential fanatics who are out to destroy the computer center. He is going to be very concerned that you will provide adequate protection for his work.

In the distribution area of your business the problem is not quite so severe. As a matter of fact, since in most cases we are talking about a small college with a single center, single staff, etc., really you don't get into this as you might if you had half a dozen different computers offering these services at half a dozen different locations.

Remember, then, that you do have a marketing problem if you are getting into selling computer time. You will have to supply your customer with a reasonably accurate estimate of his programming and operating costs for an application. You may not have had to do that in the past--I know we never did-- the registrar wants this job done, and she really doesn't care how much it costs and isn't going to get billed anyhow. So you just sort of do it and it gets done when it gets done. In addition to estimating the cost, you'll also have to set up some sort of billing procedure. This means you will have to come up with a fairly accurate evaluation of your true cost for running the computer facility. Since many of us are not businessmen, it is very, very easy to take a month's rent, divide it by 176, and say, "This is what the computer costs." That's not correct. There are overhead expenses, personnel expenses, just an infinite number of considerations, so you must be able to come up with the cost figure properly. It must not be so low that you are supporting your customers, and they are actually paying less than costs; and it must not be so high as to frighten them away. Again, if you're working as a service bureau--and please remember that the role of a service bureau, in general, is an introductory role for the small institution or small company to get into data processing--your customers will probably leave you at some point to get their own equipment, so you must make the cost of the operation something that they can accept. Since you can count on each of your customers leaving at some time, you must be constantly prospecting for new customers, bearing in mind what you have to offer and who you would like to do business with. I would warn you in this particular case about falling into the problem that so many major businesses in the country do, and that is becoming wholly dependent on one customer. If you do, and if that customer pulls up stakes, you can find yourself in a rather disastrous financial position.

Now the impact of selling computer time on your regular data-processing operations is rather considerable. First of all, I think it is vital that you establish priorities in the computer center that you might not otherwise have done. We tended to work under a FIFO arrangement prior to this, but now such things as payrolls, periodic reports, etc., have had to take precedence over some academic and administrative work. You must have the support of your college if you're going to establish these priorities, and you must have these priorities if you're going to do business. I know we got into trouble when somebody who hadn't gotten the word found out that his job had been bumped because we were doing work for somebody outside, and he got pretty upset about it.

You should probably accept the concept of a closed shop. We do not have in our installation sufficient physical facilities to be able to lock up all the data that would relate to customers so that we could just turn over the shop to students with reasonable confidence that they wouldn't get into anything. So we have just decided that it is strictly a closed-shop operation. Therefore you must accept the idea that you will need an operator. You must accept the idea that you're going to have a certain amount of your machine resources dedicated, be they disk packs, parts of core, or whatever. There are just going to be some things that are going to be unavailable to inside users. You must accept the idea that you must be a year-round operation. If your computer center closes during the summer time, particularly those that are academic-only types, very few businesses can do business with you. I don't know of any businesses that close for three months. So while you are making the decision whether to sell computer time outside, I guess the big question is whether the income will justify these "expenses." That is something you are going to have to determine within your own shop.

I will give you a guide as to where you can look for customers. We're in a very rural environment, a few thousand in the town. We have a lot of small business there, so you can look around for small business in the area, your local government--city, county government--public schools, particularly if you have some sort of terminal facility to provide, other colleges, junior colleges, community colleges. It could be that you could work out an arrangement that your college uses the computer all day and the community college comes in at night and uses it. That might be a very nice arrangement.

I will, in closing, point out the ideal customer that you're looking for. First of all, he does his own programming, so that you don't have to load your staff. Second, he buys at least a full shift, the second or third, and is fully satisfied. And third, he is an alumnus of the college, because alumni tend to think that anything they can do to help the college is really great, and boy, that's worth something!

So it's not all a bed of roses. There are implications, severe implications on the computer center operation; but it can be worth it if you set it up right.

COMMENTS ON COMPUTER CENTER POLICY

Jim McDonald
Morningside College

Several years ago we started teaching a beginning FORTRAN programming course, and, having no computer of our own, we used the University of South Dakota, which is about 45 miles away. This meant that I had to reserve the college station wagon and drive five or six students at a time so they could take programming. Later we got some organizations in town to lend us their computers, but these computers were only available at inconvenient times, and they would just as soon bump us off then as any other time. So we wound up back at the University of South Dakota. Eventually, I got tired of driving and said, "I just want to teach mathematics, and I don't want any more to do with the computer."

At that time the National Science Foundation began a program, the College Science Improvement Program, and someone pointed out to me that we might be able to get some money to get a computer of our own. Along with several other faculty members, I wrote a proposal to do this. It was during the first four months after the institution of this program, and I guess the NSF found out they were swamped with people who wanted computers. They moved into the Office of Computing Activities, and our particular draft was one of the first to go in there. Somebody made the remark yesterday, I believe, that it helps if you're first, and I think that's right. We were proposing an academic computer center; we were proposing that we buy an 1130, which we eventually did. We were visited by the NSF, and we did get a $45,000 grant to help us buy the computer.

This was still hard to do. Briefly, the installation that we got was a $94,000-$100,000 set-up. It was an ordinary 1130, single disk, 8K, 1132 printer, 1442, etc., and we bought two or three keypunches. The grant wasn't enough. It happened that at the time we were also building an addition to our science hall, and we had a Higher Education Facilities Commission grant, which also paid for a third of the contents of the building. It was logical to put the computer in that building. (This is where I would have put the computer anyway). We had $45,000 plus another $33,000, and it still wasn't enough to buy the computer. Morningside is a private church-affiliated college with about 1500 students.

One of the neighboring colleges also requested money for a computer. It was also a church-affiliated school (same church) about 25 miles away, going after this same kind of installation. Their request was turned down, and we got ours. To take up the slack, we approached them to ask if they would like to come in with us on a joint operation. It seemed a feasible thing to do at the time, and so we did this in 1968. This other college, about 1000 in enrollment, came in and bought about one-third--well, it's complicated--take $45,000 away from the $100,000, and they came in with one-third of the remainder. A computer contract was written up binding the two schools, one with the other. The $17,000 figure was one-third of the remainder after the grant was taken out, so it seemed logical to set up a one-third, two-thirds ratio of use. We have kept a log of the time of use from the beginning, and it's been roughly that. We've been open from seven in the morning until ten at night, and we have set up a schedule. Because the computer is located on the Morningside campus, the other college has been given the priority for choosing times. They've ended up taking two afternoons, two evenings, and one day--about 25 hours every week. This has worked very well, with only a few difficulties. When I've talked to other people over the years about this kind of a set-up, they've seemed amazed that the two colleges could get together.

The other mathematics professor, who taught computing at his school, was as disturbed as I had been, because he had to drive so much; but still, the facility was much better than anything that either school had ever had before. We were happy to work together. Eventually he left, and I remained.

The one-third, two-thirds ratio also involved me. In the beginning I was teaching half time and eventually I saw myself having two full-time jobs plus half of another job, so I chose to remain as Director of the computer center. The other school agreed to pick up one-third of my salary. Now I belong to two schools. Computer committees were established, one on each campus, which also met jointly. This involves two faculty members and two administrators from each campus and myself. It eventually appeared that it would be nice to have the presidents on the committees, so we made them ex officio members. We had been doing things that they didn't know about, so we put them on to see how it would work out. This has helped. They now understand our problems.

My role, then, has been to report to the academic dean at each college. I write up a weekly report estimating the percentage of time I've spent on each project. Nobody has ever questioned me on these things. It's almost like I have my own separate installation. Nobody seems to know, or sometimes care, what's going on. So we go along with our applications. In the beginning we set our usage at eighty-five percent for student time, ten percent for faculty, and five percent for administrative applications.

In the summer of 1968 the computer was installed on the Morningside campus on the 28th of August. On the 6th of September, by the time school opened, with a student's help, we were ready to do registration for our registrar. So we did have a commitment to the registrar's office. The business office had their payroll with a service bureau downtown. We encouraged them to leave it there until such time as we could afford to take it over ourselves. It's still there. I was the only member of the computer center staff. Our center, as in the other school, depended entirely on students. We hired students to do the registration and everything else. The other school eventually hired a full-time administrative programmer, assigned to the business office, and he did write a payroll package for them. They put their payroll on, but we stayed away from it.

The students who work in our center have been assigned to work with other students, with faculty members, and to supervise the computer center. We have remained on a seven a.m. to ten p.m. shift. Students operated the computer for the registrar and did a few other odd jobs for the administration.

We have had proposals from some of the other members of the Colleges of Mid-America, an eleven-college consortium, of which both schools are members, all church-related, and all smaller than we are, to see if we could do some work for them. We took on one of the other colleges right in Sioux City. This is an institution that taught the course on their campus and sent their students to the Morningside campus for hands-on time. Their students were allowed to operate the computer with our students supervising them. We had cooperated from the beginning with them by leasing the computer to them for four hours a week whenever they needed it.

We have a 1230 optical mark page reader which we use for our faculty for test analysis programs. We have worked with some of the high schools and some of the neighboring colleges in this area. We have developed some surveys of different testing and research techniques that some of our faculty as well as others, are using.

A high school has now asked us if we would do their registration procedures. We didn't really want to take it on, but they'd cancelled their other contract before they told us. We were going to be there...

As of June 1 of this year, we now have at Morningside a full-time programmer, assigned to me, to work with the administration. I see his as a multi-purpose role, too, because I would like to teach a course. I'd like for him to order the cards and paper and things like that, and allow me time to teach at least one course.

A year ago, the Sociology Department within our college--they have been a pretty good user of our computer--made a proposal to the City of Sioux City (Sioux City has about 100,000 people, the second or third largest in the state). The city of Sioux City was making a proposal to the Department of Transportation for one of the Alcohol Safety Action Project grants. There are some thirty or thirty-five of these in the United States. They wanted to use our computer to do the evaluation. Our Sociology Department wanted to do this evaluation as a subcontractor for the City of Sioux City. Their proposal was approved, with this work just beginning this summer. This is a 42-month grant to the City of Sioux City, so it's a 42-month grant to us. It's a 2.1 million dollar grant to the city, with Morningside to receive fifteen to twenty percent of that to do the evaluation. The Morningside Computer Committee managed to use, from the Sociology Department sub-contract, enough money to help purchase a 1403 Model 7, an 1133, and the Memorex 3610 disk drive. This has set us up with what we now have. We are expanding to a 24-hour day at least five days a week, and perhaps seven.

This has led to serious problems because of the contract with the other college. The contract that was set up specified certain things, and I see now the need for a lawyer. Each part of that contract can be interpreted two ways because there are two schools. If there were three, it could probably be interpreted three ways.

We have a 24-hour/day schedule, with a contract that's quite beneficial, and the other college--let's call them College A and College B--College B feels that College A has taken undue advantage of the situation, and that College B should have one-third of this $400,000. College A takes the view that the project is being done on College A's time, and that under the provisions of the contract each school has the right to allocate its share of the time to whatever use it chooses, as a precedent had been established. It was the decision of College A to use part of its share to bring in some money for the college. Each college initially had been asked to bid on the evaluation sub-contract; College A was the only college to respond.

So this spring we had a lot of new equipment coming in, and I could just see myself coming in at ten o'clock at night and finding College B using College A's equipment. I didn't know whether I would turn my back or what I'd do. College B says, "We don't have enough money to help buy this equipment but we want to use it." Well, when you have a 600-line-a-minute printer sitting beside an 1132, I would want to use it, also. And when you have five disk drives instead of one, I'd want to use that, too. So I've been caught in the middle, and have spent a great deal of time during the past year meeting with people at the colleges. As of the first of June we had handled our problems except that one night around midnight, when it came time for the ASAP project, the alcohol project, to use the equipment for the first time, they found that College B was using equipment on ASAP time. I had a knock on my door the next Monday morning, and I had to find out which contract was going to be honored. I found that one business manager was gone for the summer and the other one was at the lakes; one president was in Europe and the other was in Oregon. The president's secretary didn't seem to know who was in charge to answer my question. So I said, "Well, I'm in charge, and I'll make the decision." She said the president would be back in town on the eighth of June. I said, "I'll be in Atlanta on the eighth of June, and this is my decision. This is what we're going to do." I sent letters to the appropriate people, and then I left town.

One of the clauses in the contract says that either party may buy the other party's share with a one-year written notice, but that the other party has thirty days in which to make a counter-proposal. The faculty of College A has tried to convince the administration that now is the time to give the one-year notice, that they should buy out College B. This is probably the only reasonable approach, but problems still exist. What does College A have and what does College B have? Does College B have one-third of the total equipment? Does College B have one-third of the $100,000 or one-third of the remaining $55,000? Did NSF give the grant one-third to College B and two-thirds to College A? Where does either college obtain cash to buy out the other?

College B feels that it has one-third of the original $100,000 including the $45,000 grant from NSF. College A's business manager doesn't agree. I've done a lot of talking to business managers, too, and lawyers and everybody else. I think what will happen--I have to say that up to this point the colleges have cooperated very, very well--is that in the natural course of events, each school will require its own computing facility. Each school is now using its portion of the time to capacity. Each school has a lot of students working on the computer. Each school has several faculty members doing projects and teaching courses involving the computer. We now have FORTRAN and COBOL courses; we have taught an RPG course and a Programming Languages course using an adjunct faculty member from the local community. Long-range plans had called for the eventual termination of the contract in any case. Now we're right at the point in time, except that the way in which the termination is coming about was not part of the long-range plans.

Each school wants its own computer; College B does have a place in its new science hall in which to put one. The computer coordinator has done a good job on each campus in promoting the use of the computer. Neither school has a computer science major, minor or program of any kind. Each school has taught the use of the computer sort of like the library: a place to do research, a tool to be used in a variety of courses.

My problem as to whom I report has really been no problem, because it seems that the dean to whom I report at College A or College B is in complete cooperation with me; the computer committees have been in complete cooperation. I have no vote in either committee, and yet they have never gone against anything that I have recommended. At the same time I have never recommended anything but what I thought was needed and should be done. We have added, from the beginning, only one piece of equipment, and that was the plotter, which each school uses quite a bit.

I think the problem will be resolved, but I am now seeing the importance of having such a contract, and having a good contract between two schools.

The person that we hired to work with the administration is a former student who graduated in June, a young man who came to the college with four years of experience working as a computer technologist with the Air Force. He could already program in a couple of languages when he came on board. I had had him as a student in night classes in COBOL when he was in the Air Force. I convinced him that he should be one of my advisees and that he should major in math with business administration as a second major. So we have here about 3.5 students versed in about six languages and really good at each. I see him working now with the administration: the business manager, who wants to put on his payroll; the admissions officer, who wants a complete admissions package; an alumni director, who wants all his alumni put on; all of the deans in the college wanting various reports; and we are already completely committed to a registrar program.

We have had some people approach us with packages already developed for the 1130, complete so-called college systems to do accounts receivable, accounts payable, payroll, etc. These people say they'll do a feasibility study, they'll come in and get all these programs up for us in whatever way we want. They will train the ladies in the various offices to use these programs. The price, it appears to me, is about equal to what we're paying this man for one year. I don't think that in one year I could expect him to get all that done along with the other duties that I'd like to have him do for me. I would like to get some reaction from somebody here who has tried such packages or knows anything about people who claim that they can do such a thing for a college.

We have a decision to make, and I've attempted to get a decision from our president because this programmer is assigned to me. How will we use him? There is a job resume which I have written up. Does it look feasible? We have to establish priorities. Which administrative office will we do first? Certainly we can't do them all piecemeal, concurrently. As yet we have reached no decision. I have been given no priority, because at the same time our administration sees an opportunity to sell time or sell services, mainly to some of our sister colleges in the Consortium and to the area high schools which want service. So, do we have our man working for the college, writing its own system, or do we have him making money, so to speak--in other words, making up his salary?

I don't believe this man should have been hired to do services for people outside the college, at least in the beginning. To me, this seems an inappropriate use of his time. This is a policy that I will have to resolve in some terms right away.

SECTION V: COMPUTER SCIENCE/COMPUTER CENTER INTERFACE

Chairman: Harris Burns, Jr., Randolph-Macon College

THE FEDERAL CITY COLLEGE COMPUTER CENTER

Jesse Mayes, Federal City College

COMMENTS ON THE COMPUTER SCIENCE/COMPUTER CENTER INTERFACE

Robert Kyle, Emory University

COMPUTER SCIENCE EDUCATION IN SMALL COLLEGES; A REPORT WITH RECOMMENDATIONS

Richard Austing, University of Maryland
Gerald Engel, Hampden-Sydney College

THE FEDERAL CITY COLLEGE COMPUTER CENTER

Jesse Mayes
Federal City College

The Computer Center is established to enrich the academic programs of the College, and to enhance the education of students. The Center seeks to fulfill these purposes by formal and informal classes, laboratory sessions, disseminating information to stimulate the imaginative use of quantitative and symbolic information, and by developing and stimulating interdisciplinary curricula and seminars which relate to Computer Science. Although the computer is not used as a substitute for the teacher, it supports teaching and creative inquiry by making available to the teacher teaching tools and classroom procedures which each teacher may use to improve instruction.

The objectives of the FCC Computer Center are to:

1. Provide computer support in the advancement of Federal City College's aims in education, administration and research.

2. Make available to every student at Federal City College the appropriate instruction in computing, and required computing support in order to enhance his professional development.

3. Study the latest developments in computing applications and technology in order to insure that the most modern productive computing tools are used by Federal City College.

4. Develop a major role in the District of Columbia's program for advancing higher education through the concepts of management information processing and computing applications.

5. Provide the best computing facilities for solutions of research and development problems of the College, District of Columbia, and industry for which Federal City College may be given responsibility.

The academic workload, which constitutes sixty percent of the total system utilization, can be delineated as follows:

1. Batch processing of student jobs in a wide range of languages (FORTRAN, COBOL, RPG, PL/I and Assembler) as well as use of many IBM supplied package programs for statistical analysis or linear programming, etc.

2. Time sharing capabilities have allowed students to interact with the computer using such languages as BASIC, FORTRAN, and Assembler. The use of this facility has proven to be of significant benefit in motivating the student in introductory courses, as well as proving to be invaluable tools for upper class students in finding solutions to problems in a wide variety of academic disciplines. The use of terminals has provided in excess of three thousand student hours per month with the computer. These terminals range from an on-line registration system using TCAM to a remote job entry and spooling system using HASP, to a time sharing facility capable of supporting FORTRAN, PL/I and BASIC concurrently with other academic and administrative requirements.

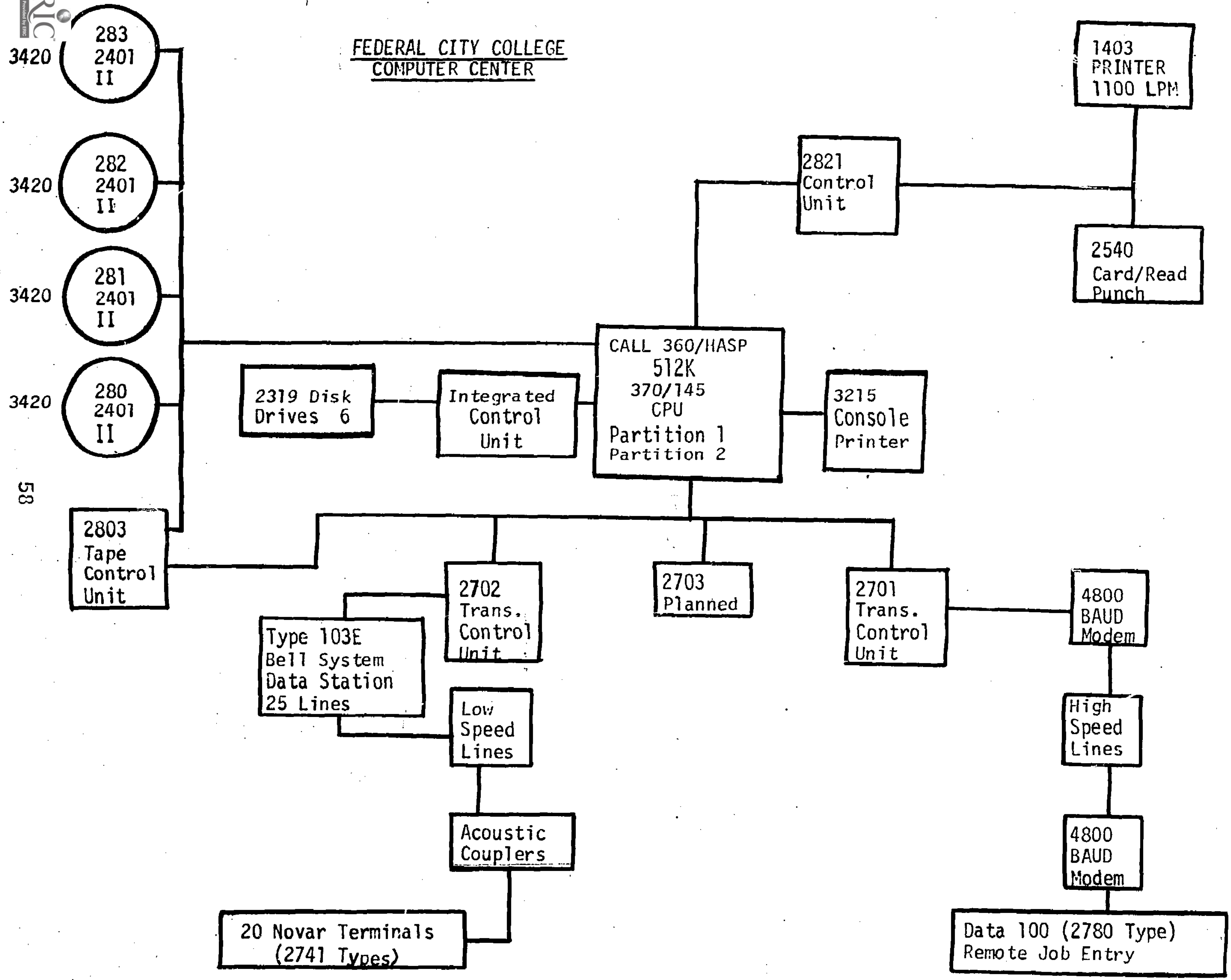
New developments:

An on-line registration system has been developed for the College. The new system allows a student to fully register in one step at a computer terminal and eliminates card pulling, accidental overrun of classes, long lines.

A new system of grade reporting has been developed for the College. On a quarterly basis, each student receives a grade report that not only shows his grades for the quarter, but shows grades for his entire college career. This enables a student to see at a glance his position and progress toward a degree. Such reports not only benefit the student, but save many man hours by eliminating the need to respond to individual requests for grade information from previous quarters. The new grade reports are prepared in triplicate and the master copy can then become an official transcript of student records after it is signed and sealed by the Registrar.

Based on the data processing requirements of the college, a comprehensive plan was put in place to meet the requirements. This plan called for installation of an IBM 370/145 with adequate core and peripherals to replace the college's IBM 360/40 computer. The IBM 370/145 was chosen for several reasons: (1) it was compatible with the existing IBM 360/40, (2) it provided three times the computational speed of the Model 40, (3) for comparable configurations that would be applicable to the college's requirements the 370/145 was less expensive than the 360/40, (4) plus a technology that will provide the college with a higher degree of systems availability, made the 370/145 a natural choice.

FEDERAL CITY COLLEGE COMPUTER CENTER



COMMENTS ON THE COMPUTER SCIENCE/COMPUTER CENTER INTERFACE

Robert Kyle
Emory University

I'd like to find out how many people in this group come from institutions of 4000 or more students. How many of you come from institutions of 2500 or less? How many come from institutions of 1000 or less students? How many of you have your own computer? Are there any of you who don't have your own computer on campus? Of those who don't have their own computer on campus, how many are using some kind of real-time system or shared system of some sort? So everyone in here is in some way already in the computer racket. Now let me get one more pair of questions and then break off from this kind of audience participation. How many of you come from institutions where your computer is used almost--not necessarily altogether, but almost--exclusively for administrative purposes? Are there any people here whose computer is used altogether for instructional or educational purposes? Okay, that gives me a good idea of where things stand. It's really quite a broad spectrum of interests and backgrounds, as one might expect. Meetings of this kind frequently announce that they are appealing to small colleges, but because small colleges can't afford to come, the medium-size colleges, who think they're small anyway because they look at their budgets and realize what they'd like them to be, are the people who make up the group. I'm pleased to see the broad spectrum of interest.

There were a good many comments earlier about curriculum. I come to you in a sense from the American Society of Information Science, a group with which I've had a good bit of involvement, and it is currently espousing a new curriculum look, inspired in part by "Curriculum '68." In the American Society for Information Science we would make the claim--incidentally, I should preface this perhaps, with Harris's charge to me to produce some nice, controversial comments and bait you into a great deal of response--that information science is the basic routine, and computer science is a facet of it. This is also true of library science and several other fields which are quite different. All these have a common interface in an area which might be called information science, and as such, the American Society for Information Science recognizes that curriculum efforts so far have all been so computer oriented, so specifically use-of-computer oriented, that they've decided to explore the possibilities of some more curriculum material. One of the interesting things to me is that your comments this morning led me to realize that there has been no consideration built into this plan, *per se*, for the smaller schools. I think that's a big mistake, and I intend to try to see that something is done about it. I don't know how far this might be carried.

I had previously thought, and still have some feeling, that the smaller schools have the same curriculum requirements that the largest schools do. They may be richer or not as rich, depending on circumstances, and perhaps have to substitute personal attention for the extra courses. But if you're going to make that statement, then you find yourself facing the issue of how you go about producing that personal attention. The computer sciences lend themselves to this in some spectacular ways, I think.

We have a program at Emory which we call the High School Fellows for Information and Computer Science. This has grown out of an activity I'll be talking about very briefly as we go along, but it involves some intense relationship with some of the high schools in the Atlanta metropolitan area, and we have invited the advisors of the schools, the people who are most involved in computer activities in these schools, to nominate one of their students--at some of the schools there are two of them--to come on our campus two days a week. They are expected to be there at least four hours a week, two hours a day. One of these hours is a very informal lecture. So far we've just made the rounds, taking the staff of the computing center, the people who are competent and interested in this kind of thing, and let them come in and talk to the Fellows for an hour. We've generally had these talks unformatted; they've just been general discussions with a fair amount of interaction. It's a little difficult to get good interaction with high-school students until they evolve some real friendship with you. They feel, apparently, quite accustomed to sitting in a classroom and being lectured to, and they're a little slow to assert themselves. They've gotten over that in our case, almost more than one can stand, but this has been a spectacularly effective program in some ways.

As I see it, the most important concept that has come out in the use of the computer for the support of other subjects, aside from computer science *per se*--the most important philosophy involved, at least--is to think of the computer as a pupil. This applies to the use of the computer in all subjects--chemistry, physics, mathematics, English, modern languages, sociology, biology, or what-have-you. In my opinion it has some potential in every one of these areas, not as a primary tool, but as an aid; and the idea of computer-as-pupil is central to its effective use. I like this idea. It's not an original one, but I surely have made a lot of mileage with it. The concept very simply is that everybody seems to learn a lot more when he's required to teach than he learns when he's required to listen, and I don't know anybody who doesn't agree with that idea. So far, everybody I've ever said this to says, "Yes, that's been my experience." Whether I'm talking about Sunday school or calculus, it's true. Well, the computer is a beautifully dumb, obedient, and demanding student. The student becomes a teacher when he's assigned the task of instructing the computer to determine the velocity with which an object will strike the floor when dropped three feet, with or without air resistance. And incidentally, when you use the computer, there's no reason to leave out air resistance as probably everybody in this room did when he took physics. It applies just as well when you're talking about ion mobility or anything else. The idea of the computer as a pupil, in my opinion, is the one that really ought to be pushed among all the ancillary uses of the computer.

Now, what happened to these High School Fellows is a beautiful example of this, but in their case it is occurring in computer science, where it does not occur so naturally. We have to operate on a low overhead with these boys. As a matter of fact, we haven't budgeted one penny for them, although we've given them an awful lot of free computer time. We allow them to use the terminal in my building, which is separate from the main computing center, but they would overflow into every terminal on campus if we let them. We kept the group small; we had no idea how many we could tolerate in this kind of category, because we knew they'd be precocious, inquisitive, and demanding. A great deal of pressure has been placed on them to be self-effacing. If he sees anybody on the staff who looks like he wants to use a terminal, the Fellow is supposed to stand up and say, "May I log off for you?" Practically--it's almost that way. Anyway they're good about it, and they're really highly liked by the majority of the staff.

I think this concept of "Come in and do it yourself" might be a very significant feature for any of the students that can be turned on. Now about ten to twenty percent of the average student body is likely to get turned on if they're given fairly free access to a computer with enough tools to learn how to use it. To get turned on for a year or so is all it takes to become really good, in my opinion. That doesn't necessarily mean they'll be computer scientists eventually. They may drop from it after that time. But they will have fitted themselves well for their role in a society that's so heavily computer-based if they become intensely interested for that period of time.

If you have ten or twenty percent of a student body that is intensely interested, the spillover onto the rest of the students is tremendous. The spillover onto the faculty is also tremendous, which is one of the places where it is very difficult to get it in my experience.

Now to get to the more germane parts of the conversation--that was the preamble, and I'll take less time on the germane than I did on the preamble, I believe--about two years ago we got the idea that we were hard-pressed financially. A lot of people had trouble looking at Emory, which I believe is the nineteenth best-endowed university in the country, as being hard-pressed financially, but we, like everybody else, find that our income doesn't meet our expenses. There are a lot of problems. We saw ourselves as needing additional computer capacity. A lot of people are horrified when we talk about that; we've got eighteen computers on campus for instance. It frightens even me! We're not really a big show in computing, or anything like that; we just have a lot of research that uses them. But this gives us a very strong place, of course, and many academic departments are very strongly involved in computer instruction. We hit on the idea that perhaps some of our less fortunate sister institutions in the immediate area might wish to participate in some of our gifts in such a way that we might mutually benefit. We hoped that we would be able to charge them an awful lot less than they would have to pay elsewhere. We could give them what they would choose to have, from student involvement on our campus, to terminal access, to remote job entry ports--just about anything they wanted. We visualized a state network of private colleges.

At the time we weren't fully aware of how far the public system had gotten in its network and that it was available to private schools. It seems that the public school people sort of forgot to mention to Emory that this was going on. We've wondered at times if there might have been some reasons for that. For awhile the University System shared three CPU's in a network: Georgia State, Georgia Tech, and the University of Georgia, and a lot of serious consideration was given in the past year to the possibility that these three schools and Emory would share these resources.

It has turned out that many of the private schools have felt a real reluctance about joining the university network, and they tell me it is partly because they have felt generally swallowed up. When they get into the system they do not have sufficient usage, sufficient voice to be sure of keeping a total role in the system. Nobody is out to get them, nobody is taking advantage of them. The University System people are just as nice as they can be about that kind of thing, but they just sort of forget about the school of a thousand students that doesn't have any facilities of its own and is strictly using the telephone system to get into the main net. This seems to have happened in some other areas, and some of the schools have expressed a concern about it. I do not know how valid this concern is. But there was an interest in using Emory as a "big brother", where we did have a staff and did have sufficient background to speak on an equal basis with any of the other universities in the system. This served as the basis for the concept of the formation of a private university network. This has not been a highly successful concept. Except for Atlanta University where there are several of our terminals, a very modest amount is involved in this organization. Some of the other schools that make up the Atlanta University complex are expected to join in the coming year, so there will be a fair involvement there.

We tried to find what mechanism could lead us to a base for these other schools. First of all, we established what we considered to be a reasonable contract. We allowed the schools to have access to ports to our system. We use what was RCA, a Univac Series 70/46 for our real-time work and expect to expand the service in the immediate future. We offered a port to this system using BASIC only, although we also allow a rather high-level text processor, EDP, an RCA editor. This is, in my opinion, perhaps the best editor available, at least the best I've had the opportunity to explore. We do allow them to use a disk for string operations--text processing--and we permit also some of the other minor activities to be used. We allow this contract for $600 including the terminal. If they are remotely located there is, unfortunately, a communications-line cost which frequently equals the computer charges. The $600 allows unlimited use of the system from eight in the morning to eleven-thirty in the evening, and we also try to give them some help in getting started. Some of the schools we talked to had no computers. They are not as sophisticated as every one of you. There are others that have been experimenting with it or dabbling with it. We see this as a valid activity for even more sophisticated schools. We have an alternate contract at $800 which allows them the entire use of the whole system. Both contracts provide some disk storage: fifty of our PAM pages, which is 2048 characters, for the $600 contract, and three times that much for the larger one. So they get a fair amount of additional storage, enough to make it useful and to get them away from paper tape and this sort of thing.

We decided that the primary thing that most of the schools we were talking to needed was financial support that they just couldn't come up with themselves. The funds at the federal level at that stage were pretty well dried up for this kind of activity. We conceived another plan which I think has a great deal of merit, and I would like to propose it for other schools to pick up, with or without cooperative features to some central system involved. We recognized that in the boondocks of Georgia, it would probably be ten to twenty years before any of the people in the high schools or grammar schools have any access to a computer if activities wended their natural course. We suggested that a very natural affiliation would be between the many colleges over the state and the public schools. When you look at a map of all the public and private colleges in any state, it turns out that they do follow population, they do cover the geography of the state, and there are not long transmission-line requirements. Frequently there are no toll line requirements for reaching the nearest college from any place in most states. This means, to me, that the small colleges could become managers of computer systems to serve the public schools. Many of the smaller public schools can't afford a system of their own, and the private schools almost uniformly can't. There are three in Atlanta that have done something along this line, and that's all, out of forty-five private schools in Atlanta. I'm talking about pre-college private schools at this point. The only one that's really done a major job received an unbelievable grant. Most of the secondary schools don't get unbelievable grants; this one did and has done a beautiful job with it. The others have jury-rigged and been hit-and-miss in their approach, and they have needed some other kind of support. It appears to me that many colleges recruit a fair proportion of their student bodies from the immediate area. Many universities and colleges feel the need for public relations with people in their general vicinty. I think there is a great deal of merit in this concept. I think there is, economically, a great deal of potential in it.

The small public school systems and the private schools can't even afford computers for their administrative needs. The role of the computer in the classroom as an educational aid in the high schools and grammer schools is up for grabs in two respects: One of them is that there aren't many people who really understand what's going on and what the possibilities are. And, second, even those who do, don't really have good concrete data to try to justify a system. The primary advantage of the computer in these locations is the impact it has on the student outside of the course in which he studies the computer, on his attitude toward school and toward homework. There are potentials here that we don't even have time to touch on today. But I see these things as being among the most important activities that a school could take on, and having the possibility of offering economic potential.

We would like to see a network of systems in which some of the universities and colleges located around the state might have their own computer and provide a realtime service to the high schools; some others might physically transport some of the students into their schools on a fee basis to use some of their resources at certain times of the day. We see other possibilities for minicomputers, compiling in a central location but doing all of their running on a strictly local basis without having to pay for communications lines. There are myriad configurations for this sort of thing, myriad orientations, many of which do have some real potential. In Atlanta in the past two weeks we have organized something called the School Computing Council. Its purpose is to evaluate and implement these applications. We hope this will grow into something viable and important in Atlanta, and possibly other organizations might find this useful to consider as a concept for themselves.

COMPUTER SCIENCE EDUCATION IN SMALL COLLEGES - A REPORT WITH RECOMMENDATIONS*

Richard H. Austing, University of Maryland
Gerald L. Engel, Hampden-Sydney College**

The Subcommittee on Small College Programs of the Committee on Curriculum in Computer Science (C³S), of the Association for Computing Machinery was appointed in 1969 to consider the unique problems of smaller colleges and universities, and to make recommendations regarding programs at such schools.

Through the efforts of this group, with support from the National Science Foundation, an institute was held, in the summer of 1971 at Purdue University, on undergraduate computer science education. At this institute fifty-three instructors from smaller colleges and universities received instruction regarding the teaching of courses in discrete structures, programming languages, data structures, and operating systems. In the process of evaluating the results of this institute, it became apparent to the committee that recommendations regarding programs in computer science specifically directed to the smaller schools were needed. It is the purpose of this report to supply these recommendations.

In no way does the material in this report represent a major program in computer science; rather a program is recommended for those schools with limited resources, but with an interest, enthusiasm, and desire for some course offerings. Those institutions interested in, and with the resources necessary for a major program, should refer to the existing reports of C³S and other curriculum studies.

The Program

Included in this report are the descriptions of four courses. Though the attachment of specific names has been avoided, the courses correspond roughly to the areas of algorithms and programming (Course 1), application of computers and their impact on society (Course 2), machine and systems organization (Course 3), and file and data organization (Course 4).

Though these courses in a real sense represent a coherent program, they are so structured as to allow a student with limited objectives and limited time to pick and choose those parts most relevant to his needs. It is anticipated that these courses can be supplemented with independent and directed study courses for those students desiring further work.

The main constraint in a program in a small school is staff size, and the related lack of multiple sections. Most schools have no more than one full time equivalent faculty member available to teach computer science, and in most cases the same course and section must serve students with a variety of backgrounds and objectives.

Course 1 is the introduction, which in most cases, gives a student his first experience in computing. This is accomplished primarily by the presentation of a higher level programming language. Course 1 is a prerequisite to the other three courses. Course 2 expands on Course 1 by giving the student further programming experience. In addition the student is introduced to a variety of applications of computers, and the effects that these applications will have on the individual and on society. In Course 3, the student gains familiarity with various aspects of computer systems, and how the parts of such systems interact. Finally, in Course 4 the concepts and applications of data representation and organization are considered.

Three of the courses (Courses 1, 3, and 4) correspond to courses in "Curriculum '68," in basic content, however, there is a good deal of difference in structure and emphasis. In order to allow as many students as possible to take the courses, the prerequisite structure is held to a minimum. Also, in order to provide a more general background, the Courses (especially courses 3 and 4) are more concerned with concepts than with details of a particular system, or extensive programming exercises. For example, in Course 3 no particular assembler would be studied, but rather the general concept and vocabulary of computer systems would be presented. In this way a student, anticipating a career in business management could equip himself with the tools to select a computer system, without having to bury himself in the details of a particular system.

---

*The work reported here was supported in part by National Science Foundation grant GJ-1177 to the Association for Computing Machinery.

**On leave at Computer Science Department, The Pennsylvania State University.

Course 2 does not have an equivalent in "Curriculum '68." This course in applications would in most cases be the natural sequel to the introductory programming course. It combines further experience in programming with a survey of application areas. Though programming would be an integral part of the course, something of the overall descriptive nature of the program would be involved. Where possible and appropriate, the students would be expected to use programs and data bases that are available; thus, for example, if the class were studying simulation, it would be appropriate for the student to gain experience by playing some computer based game, and possibly to study some of the techniques involved in writing appropriate programs, but not necessarily to write game playing programs.

By the implementation of this program, instruction would be available to all students on campus, at the level of being able to communicate intelligently with a computer. In addition, advanced instruction would be readily accessible. For the student anticipating a career in computing, or considering application for graduate work in computer science, several approaches are possible. Independent study courses can provide introductions to certain topics; such courses in assembly language programming, programming languages, or even some large scale programming project would be appropriate. Also, since we are dealing with small schools, cooperation with other departments can be anticipated. Through this interdepartmental cooperation, certain courses can be modified to serve the student anticipating graduate work in computer science. Such a student should be advised to follow a mathematics curriculum, and could anticipate taking at least a computer oriented course in numerical analysis, and a course in abstract algebra that would emphasize computer applications. Finally, with the general introduction of computers in the undergraduate curricula as is documented by the Conferences on Computers in the Undergraduate Curricula (1970 at the University of Iowa, 1971 at Dartmouth College), it seems reasonable to anticipate that an interested student can select several courses from various disciplines that make significant use of computers.

Implementation

One of the purposes of this program is to ensure its implementation with a minimal staff. Obviously, computing equipment must also be considered, and since most small schools are working und " a small budget for computer services, the course structure reflects the fact that extensive computer power will probably not be available on campus. The courses recommended require that the students have access to a computer which has a higher level programming language for student use. Only one higher level language is required, inasmuch as every installation satisfies that requirement. If additional languages are available, their use might be appropriate in one or more courses. Whether the computer is a small stand-alone or one or more terminals makes little difference. The important requirement is that the students have easy access to the equipment and good service, both in terms of turn-around time and debugging facilities.

As important as the computer science course structure is, the most important area of computing at a small school is the service area. The cost of computing on campus, both in terms of equipment and personnel, can only be justified if computing services are used on a campus-wide basis. To achieve this, an excellent development of Course 1 is necessary, as well as the development of a community of computer users on campus. For schools that are not already involved in such programs, the first efforts must be made in these directions. In fact, the first course, and the development of users on campus, should be the first responsibility of the faculty member in charge of the development of computing, and the introduction of the additional course work should take place after these aspects of the program are completed.

The program requires one full-time instructor. In most cases, Course 1 and Course 2 would be offered each semester, while Course 3 and Course 4 would be offered once each year. It is common practice in small schools to have the computer science faculty and computer center staff one and the same. It is clear that the demands of this program (at least 9 hours per semester) make this situation impossible. Thus the instructional staff and computer center staff should be separate; however, it would be possible (and probably desirable) for the instructor to have some responsibilities in the area of user services.

Another common practice in small schools is to take a faculty member from a department that is a computer user, and assign him responsibility for computer science instruction. Such a practice often leads to the courses not being computer science but rather applications of computers. Whenever possible this should be avoided, but if it is necessary, the instructional material should be clearly separated from any other department of the school.

It is well to note that the present market situation places a small school in an excellent position to hire a computer scientist. Where possible this should be done, at least to the extent of bringing in the individual responsible for the implementation of the program. Where this cannot be done, a commitment should be made to allow an existing faculty member to develop himself in computer science education. Summer programs for this purpose are not plentiful, and doing such work in the normal environment of teaching and other responsibilities at the small school is close to impossible. Thus, where an existing faculty member is asked to be responsible for the program, it is strongly recommended that this faculty member be granted a year's leave to work and gain experience in a computer science department. It is also recommended that universities with the facilities develop programs that will assist these faculty members in achieving their objective.

As with any program, the usual supporting faciliites of the college are necessary. Though no great amount of specialized material is expected, it should be recognized that there will be a need for a rather large initial expenditure in the area of library materials, both books and periodicals. To provide a starting point for the development of a collection, a library list is included in this report.

Courses

There is much evidence that some exposure to computers should be an essential part of every college student's education. Many students will become users in their chosen occupations. Included in this group would be teachers, managers, researchers, and programmers who will need the computers as a tool. Other students will become directly involved in computer education and the computer industry. All students will be affected by the use of computers in our society.

Minimally, students should acquire some understanding of the implications of the computer impact on individuals, organizations, and society. One way in which an academic institution can do this is to offer a survey type course in computers and society. However, there are some inherent difficulties with such an approach, particularly in schools which have no more than one or two faculty members in the computer science area. The breadth and amount of knowledge needed to give a worthwhile course of this type almost precludes it being offered by any one person. Developments and applications span such a wide range of areas that faculty from a variety of fields would need to be used. The course then might take on the flavor of a lecture series in which students would be presented a great deal of information but almost no feeling about what a computer is or how it should be used.

A better approach, as well as a more practical one in terms of faculty utilization, would consist of teaching fundamentals of computing in a first course and allowing students the option of acquiring additional knowledge through their own reading, on-the-job training or further course work in computer science or other disciplines. The first course described below follows this approach. It plays the role of a beginning course and the prerequisite course to each of the other three courses described. The latter three courses are designed not to be sequential. However, the most desirable path through them for students taking all of them would be in the order presented.

There is an intended over-lap in material among the four courses. Some ideas are worth repeating at different levels. Also, the same problem or concept can be enhanced by looking at it from different points of view or by bringing different material to bear on it.

Very few matters related to courses or curriculum are generally agreed upon among computer scientists. The question of what language to teach in a first course is no exception. Although there appears to be general agreement that a higher level language should be presented before an assembly language, there is substantial difference of opinion regarding the specific language to use. BASIC, FORTRAN, a hypothetical language, and PL/1, to name a few, each have a band of advocates. FORTRAN IV is still the most widely used general purpose language and is the most easily transferable from computer to computer. Despite its shortcomings, FORTRAN IV would seem to be the most useful for the greatest number of students, and is the language recommended for the first couse. Pl/1, if it is available, could be chosen in place of FORTRAN, particularly because its capabilities for nonnumeric applications make it useful in courses 2 and 4. If strong reasons compel a different choice of language, some modifications might be necessary in course topics or approach. The introduction of a second language (e.g. ALGOL, APL, SNOBOL 4) is not recommended; it greatly decreases the programming experience and competence the student acquires in the first course.

COURSE I (3 CREDITS)

INTRODUCTION. This is a first course which emphasizes good programming techniques in a higher level language. No computing background is assumed. Upon completion of this course, the student should: a) have practical experience in programming, including modularization of both a problem and a program for its solution, debugging, implementation of basic data structures such as lists, and use of "canned" programs; b) know basic characterization of computer organization; c) be able to distinguish among program assembly, loading, compilation and execution, including some of the kinds of programming errors that can occur at each stage; d) know the details of the language and have a basic idea of the relation of its statements to machine code.

The list of topics for this course does not differ substantially from the topics included in the outline of course B1 in "Curriculum '68," however a shift in emphasis is recommended. Course B1 stressed the notions of algorithm, problem analysis, and the formulation of algorithms for problem solution. Learning a language, practice in its use, and concepts of computer organization were also emphasized, but mainly as the means to obtain the actual solution of the problem. Unfortunately, no texts have appeared which have achieved the goal of presenting the subject of problem solving in an effective way (several books by Polya might be considered exceptions to this statement but they are not of the algorithmic orientation specified in course B1). Judging from the great variety found in introductory computing courses few, if any, teachers have been able to achieve the goal. It is not an easy problem to solve.

On the other hand, it is possible to teach programming techniques with the aid of a language manual and, possibly, one of the existing texts. The latter books could be used as a source of problems at least, and, in some cases, to supplement discussions of appropriate programming techniques applied to specific classes of problems. By concentrating on programming, the instructor is better able to teach a language, put it in proper perspective with computer organizations and systems, develop good programming practices (including coding, debugging and documentation), and motivate the need for algorithms in the solution process. Students should be required to use subprograms extensively (both their own and ones that are provided); this, in turn would encourage at least one good problem solving technique - breaking up a problem into modules.

An important benefit to the general approach suggested here is that the course is more easily defensible as a service course. Students could be urged to find problems in their own field of interest which they would program as course projects. Duplication of first courses for different groups of students could be minimized, and, possibly avoided entirely. For the first few semesters it might be difficult to obtain reasonable problems from a variety of areas but, as more faculty members become users, they will become a source of good problems. In addition, a collection of (possibly large) data bases and subprograms can be accumulated and used as files to be referenced by student programs. The degree of success achieved by the computer center in developing a community of computer users has a significant influence here. As a result, some very interesting and nontrivial problems can be considered both in this course and in Course 2.

The course should be scheduled to allow for laboratory-like sessions for small groups of students. An instructor may want to scatter these sessions throughout the semester, or bunch them at the beginning of the course and let the students program on a more individual basis toward the end of the course. Whether or not the laboratory sessions should be regularly scheduled is a matter that is best decided by the instructor and/or the department.

CATALOG DESCRIPTION - A first course in programming using the language FORTRAN. Introductory concepts of computer organization and systems. Programming projects, including at least one from the student's field of interest.

OUTLINE. Material in this course need not be presented in the sequence used in this outline. Some topics (e.g., computer organization) should be distributed throughout the course with increasing degrees of detail.

1. Overview of a computer. Hardware components; how programs are executed. (5%)

2. Overview of problem solving process, beginning with the problem statement and ending with verification of the correct computer solution. (5%)

3. Introduction to the specific computer environment in which the student will work. Information needed by the student to interact with the computer in this course. (5%)

4. Language details. Components and types of assignment, control, and specification statements; data representation and structures; storage allocation; I/O; subprograms; local and global variables; common and equivalence statements. (30%)

5. Programming techniques. Segmentation; comments and other documentation; debugging; library subroutines. (15%)

6. Simple data structures and list processing. Pointers; structures such as strings, stacks, linear and circular lists. (10%)

7. Limitations of FORTRAN. Non-numeric programming; recursion. (5%)

8. Computer organization and systems. More detailed presentation of hardware and systems software, including registers, instruction codes, addressing, assembler, loader, compiler, and characteristics of components; peripheral units; past, present and future developments. (20%)

9. Examinations. (5%)

TEXTS. A language manual, either the manfacturer's or one of the numerous manuals and primers that are available, should be used. Also, any local documentation concerning the installation's computer and/or system should be readily available. No current book covers the material as presented in the outline, but parts of many books could be used as source material or student reference. For example, the following references are pertinent: 1, 2, 3, 4, 10, 13, 16, 21, 23, 27, 31, 33, 41, 62, 67, 71, 72, 74, 75, and 80 h-k.

COURSE 2

INTRODUCTION. This course emphasizes the use of computers in a variety of problem areas. It is an applications oriented course which should give the student concrete experience in solving representative problems of a practical nature. As in Course 1, large data bases can be established as experience is gained in teaching the course. Discussion of problems and problem areas should include algorithms, application of techniques from Course 1, and social implications. New concepts and tools (e.g. complex data structures, tree search techniques, sorting methods) can be introduced as required in the context of specific problems and the need for additional tools, including different kinds of languages, can be motivated. Occasionally, it might be feasible to invite a faculty member from another department or university or a local businessman to supplement material on a topic. Student assignments should vary, both in depth and in subject areas. In particular, a student who has completed Course 3 or 4 should be expected to use different techniques and solve larger or more difficult problems than a student who had completed only Course 1. Students should be encouraged to discover and solve problems in their own areas of interest.

Because students in this course have completed a programming course, no discussion should be necessary on such topics as what a computer is and how it works, number representation, flowcharts and other elementary matters included in a computer appreciation-type course. However, a discussion of various systems (timesharing, batch, etc.) should be included so that students are aware of the kinds of computer environments in which problems are solved.

The instructor should pose a suitably difficult problem in a real context, indicate possible approaches to its solution, break it up into smaller problems, discuss appropriate algorithms, introduce whatever new topics pertain to the problem, and let the student write a program to obtain the solution. (If an entire problem is too difficult to solve in this way, one or more subproblems can be identified and handled as described. More advanced methods can be indicated when appropriate and the student can be directed to appropriate references. Social and historical implications can be discussed at various stages of the solution process. As the course progresses, students should be expected to do more analysis and algorithm writing than specified above. The desired effects are that the student becomes acquainted with the computer's impact in a number of areas, is exposed to concepts and methods applicable to different kinds of problems, and gains practical experience in solving problems.

CATALOG DESCRIPTION. Prerequisite, Course 1. Social implications. Computer applications in areas such as file management, gaming, CAI, simulation and modeling. Problem solving with emphasis on analysis, formulation of algorithms, and programming. Projects chosen from various applications areas including student's area of interest.

OUTLINE. The selection and ordering of topics is highly dependent on the local situation. The topics are listed separately but should be combined as much as possible during discussions of specific problems. Problems and projects should have a practical flavor and should use a variety of computer oriented techniques and concepts. Attention should be given to the kind of technique that applies to a particular class of problems but not to other classes of problems. Each problem should be discussed in such a way that the student is aware of its relation to a real world context and sees the computer as a natural tool in the solution process.

1. Computer systems. Batch and interactive; real time; information management; networks. Description of each system, how if differs from the others and kinds of applicatons for which each system is best suited. (15%)

2. Large data bases. Their establishment and use; data definition and structures. (10%)

3. Errors. Types; effects; handling them. (5%)

4. Social implications. Human-machine interface, privacy; moral and legal issues. (15%)

5. Future social impact. Checkless society; CAI; national data bank. (10%)

6. Languages. Business oriented; list processing; simulation; string and symbol manipulation. Brief exposition of characteristics which make these languages appropriate for particular classes of problems. (15%)

7. Concepts and techniques used in solving problems from applications areas such as CAI, data management, gaming, information retrieval, and simulation. (25%)

8. Discussion of completed projects and/or examinations. (5%)

TEXTS. The underlined references cited below could serve as basic texts for this course. Many books and magazine articles could provide useful supplementary material either for class use or for student or teacher reference. Only a sampling of the available material is included in the Library List: 2, 7, 8, 11, 15, 16, 17, 18, 33, 35, 43, 55, 56, 58, 60, 61, 66, 68, 70, 71, 73, 79a, 79b, and 80h-k.

COURSE 3

INTRODUCTION. This course emphasizes the relationships between computer organization (hardware) and systems (software). Each component's organization should be discussed and its features should be related to the implementation of programming language features and to assembly language instructions. Whenever possible, explanations should be included as to why specific hardware features are better suited than others to certain types of problems or environments (e.g. real time computing, interactive systems, data processing, scientific applications), and how this could affect selection of components. The effects of adding or changing components should be viewed with respect to costs, capabilities, and software. Minicomputers should be discussed both as stand-alone computers and as components of larger systems.

Programming in assembly language should not be taught as such. However, students should be exposed to the use of macros and microprogramming. They should acquire a basic understanding of monitors, interrupts, addressing, program control, as well as implementation of arrays, stacks and hash tables. In snort, they should become familiar with assembly language concepts but in relation to their use in the total computer environment rather than through extensive programming. The need for assembly language programming experience is no longer great enough to argue that most students should have it. For those students who become interested in it, a special study course can be provided. With the background acquired in Course 3, a student should be able to gain programming experience without much additional guidance.

CATALOG DESCRIPTION. Prerequisite, Course 1. Relationships among computer components, structures and systems. Hardware features, costs, capabilities, and selection. Assembly language concepts and implementation.

OUTLINE. Because this course is, at least to some extent, dependent on the specific computer available, the selection, ordering, and depth of coverage of topics will vary from institution to institution.

1. Processor. Arithmetic and control functions; relationships of features to language features; data handling; addressing. (20%)

2. Memory. Various types; cost, capabilities, and functions of each type; direct, random and sequential access; implementation of arrays, stacks, and hash tables. (20%)

3. I/O. Types, costs, and capabilities of units and media; control; channels; interrupts. (20%)

4. Communication among components. Effects of changing configurations; interactive and real-time systems. (10%)

5. Minicomputers. Capabilities as stand-alone computers; components of larger systems; costs. (10%)

6. Assembly language concepts. Instructions and their relations to components included above; macros, microprogramming. (15%)

7. Examinations. (5%)

TEXTS. No available text is suitable for this course. Material can be drawn from the following references and from manufacturer's manuals: 4, 5, 6, 14, 17, 25, 26, 28, 29, 30, 32, 34, 37, 38, 39, 40, 42, 44, 45, 46, 47, 52, 55, 55, 57, 63, 64, 65, 68, 69, 76, 78, 80a, and 80c.

COURSE 4

INTRODUCTION. This is a course in file organization and manipulation. It stresses concepts, data structures, and algorithms used in the solution of nonnumerical problems. Proper motivation for each should be given; an encyclopedia approach is not intended. Whenever several methods for achieving the same result are discussed (e.g., sorting or searching algorithms), comparative evaluations should be included. Differences between using core only and core plus auxiliary memory for various applications should be pointed out. If appropriate hardware is available, students should be assigned programming projects that require performing operations on large data bases and that require manipulating records on auxiliary memory devices. Immediate sources of problems are in the areas of mailing lists, registration, scheduling, student records, library automation. If a suitable language for list processing applications is available, it could be taught and used in part of the course. Otherwise, characteristics of languages for this purpose should be given.

CATALOG DESCRIPTION. Prerequisite, Course 1. Data structures, concepts and algorithms used in the solution of nonnumerical problems. Applications to data management systems, file organization, information retrieval, list processing, programming languages.

OUTLINE. Neither mathematical applications nor mathematical properties of structures is included in this outline. They could become part of the course if students have sufficient background. Although some of the topics are discussed in Courses 1, 2, and 3, only the material in Course 1 is assumed.

1. Stacks, queues, arrays, lists. Structures; algorithms for manipulating, storage allocation and maintenance; applications. (25%)

2. Languages for list processing. Features of one or more languages (e.g. LISP, L6, PL/1). (5%)

3. Trees. Binary; threaded; traversal schemes; storage representation; applications. (25%)

4. Hash coding. Addressing; collisions; applications to symbol tables. (10%)

5. Searching and sorting. Comparison and evaluation of methods; techniques for use with auxiliary memory devices; applications. (15%)

6. Complex structures. Hierarchical; indexed sequential; inverted list; multilinked; applications to large information systems including case studies with illustrations of why they might not work. (15%)

7. Examinations. (5%)

TEXTS. A text for this course could be chosen from the underlined items included in the following list. However, the text would have to be supplemented with material from other references.

9, 12, 19, 20, 22, 25, 27, 30, <u>36</u>, 40, 43, 46, <u>48</u>, 50, 53, <u>69</u>, 80a, and 80c.

The four courses described above are designed to service a broad segment of the undergraduate student body with an extremely limited number of faculty members, possibly one. Students should also have the opportunity to take computer oriented courses in their own departments. The number of possible courses in this category is too great to try to list. Instead, we will recommend additional courses for the student who is seriously interested in computer science, whether or not that student intends to pursue a graduate degree program in the field.

Each of the following specific courses could be given for special study to one or a few students or as a regular course if the demand is great enough and an instructor is available. Other topics could be included, but might not be possible to implement in a practical way unless access to a large computer was available.

a. Assembly Language Programming. This course would enable a student interested in systems to apply the concepts learned in Course 3 and would provide a means to become introduced to systems programming. Manufacturer's manuals would initially serve as texts. The COSINE Committee's report, "An Undergraduate Course on Operating Systems Principles" (June 1971) provides a number of ideas for possible topics and references after the student acquires some programming experience.

b. Structure of Programming Languages. This course would include an introduction to grammars, languages they generate, scanners, recognizers and other topics as time allows. A suitable text would be <u>Compiler Construction for Digital Computers</u> by David Gries. Supplementary material could be taken from "Ten Mini-Languages" by H. F. Ledgard or <u>A Comparative Study of Programming Languages</u> by B. Higman. Also the features of languages such as ALGOL and SNOBOL4 could be studied.

c. Programming Languages. If any language other than those included in courses is available, a special study programming course may be appropriate. Such a course might carry only one credit and it might be best given as a month-long course in schools on the 4-1-4 system.

A student interested in Computer Science should be advised to follow a mathematics curriculum. The Mathematics Department should be urged, if necessary, to offer computer oriented courses. These could begin with a computer oriented calculus sequence. Abstract algebra, differential equations, linear algebra, logic, numerical analysis, probability and statistics could also utilize the computer and would be valuable courses for graduate school preparation.

LIBRARY LIST

The following list is not exhaustive. No attempt was made to compile a list of all books on any specific topic. Certain areas are omitted entirely; namely, programming language manuals and books primarily oriented toward use in other disciplines (such as numerical methods, computers and music, and programming for the behavioral sciences).

1. Arden, B. W. An Introduction to Digital Computing. Addison-Wesley, 1963.

2. Barrodale, I., Roberts, F., and Ehle, B. Elementary Computer Applications. John Wiley, 1971.

3. Barron, D. W. Recursive Techniques in Programming. American Elsevier, 1968.

4. Barron, D. W. Assemblers and Loaders. American Elsevier, 1969.

5. Beizer, B. The Architecture and Engineering of Digital Computer Complexes. Plenum Press, 1971.

6. Bell, C. G. and Newell, A. Computer Structures: Readings and Examples. McGraw-Hill, 1971.

7. Bemer, R. M. (Ed.) Computers and Crisis. ACM, Inc., 1971.

8. Benice, D. D. (Ed.) Computer Selections. McGraw-Hill, 1971.

9. Berztiss, A. T. Data Structures: Theory and Practice. Academic Press, 1971.

10. Brooks, F. and Iverson, K. Automatic Data Processing. John Wiley, 1963.

11. Clark, F. Information Processing. Prentice-Hall, 1970.

12. Clark, K. W. Use of Files. American Elsevier, In press.

13. Cole, R. W. Introduction to Computing. McGraw-Hill, 1969.

14. Cuttle, G. and Robinson, P. B. (Eds.) Executive Programs and Operating Systems. American Elsevier, 1970.

15. Davenport, W. P. Modern Data Communications. Hayden, 1971.

16. Desmonde, W. H. Computers and Their Uses. Prentice-Hall, 1971.

17. Dippel, G. and House, W. C. Information Systems. Scott, Foresman, 1969.

18. Feigenbaum, E. A. and Feldman, J. (Eds.) Computers and Thought. McGraw-Hill, 1963.

19. Flores, I. Sorting. Prentice-Hall, 1969.

20. Flores, I. Data Structures and Management. Prentice-Hall, 1970.

21. Forsythe, A. I., Keenan, T. A., Organick, E. I., and Stenberg, W. Computer Science: A First Course. John Wiley, 1969.

22. Foster, J. M. List Processing. American Elsevier, 1967.

23. Galler, B. A. The Language of Computers. McGraw-Hill, 1962.

24. Galler, B. A. and Perlis, A. J. A View of Programming Languages. Addison-Wesley, 1970.

25. Gauthier, R. and Ponto, S. Designing Systems Programs. Prentice-Hall, 1970.

26. Gear, C. W. Computer Organization and Programming. McGraw-Hill, 1969.

27. Gear, C. W. Introduction to Computer Science. Science Research Associates, In press.

28. Genuys, F. (Ed.) Programming Languages. Academic Press, 1968.

29. Gordon, G. System Simulation. Prentice-Hall, 1969.

30. Gries, D. Compiler Construction for Digital Computers. John Wiley, 1971.

31. Gruenberger, F. Computing: An Introduction. Harcourt, Brace and World, 1969.

32. Gruenberger, F. Computing: A Second Course. Canfield Press, 1971.

33. Gruenberger, F. and Jaffray, G. Problems for Computer Solution. John Wiley, 1965.

34. Gachwind, H. W. Design of Digital Computers, An Introduction. Springer-Verlag, 1970.

35. Hamming, R. W. Computers and Society. McGraw-Hill, 1972.

36. Harrison, M. C. Data Structures and Programming. Courant Institute of Mathematical Sciences, New York University, 1970.

37. Hassitt, A. Computer Programming and Computer Systems. Academic Press, 1967.

38. Hellerman, H. Digital Computer System Principles. McGraw-Hill, 1967.

39. Higman, B. A Comparative Study of Programming Languages. American Elsevier, 1967.

40. Hopgood, F. R. A. Compiling Techniques. American Elsevier, 1969.

41. Hull, T. E. and Day, D. D. F. Computers and Problem Solving. Addison-Wesley, (Canada) 1970.

42. Husson, S. Microprogramming: Principles and Practice. Prentice-Hall, 1970.

43. IFIP. File Organization. Selected papers from File 68 - an I.A.G. Conference. Swets and Zeitlinger N. V. (Amsterdam), 1969.

44. Iliffe, J. K. Basic Machine Principles. American Elsevier, 1968.

45. Iverson, K. A Programming Language. John Wiley, 1962.

46. Johnson, L. R. System Structure in Data, Programs and Computers. Prentice-Hall, 1970.

47. Katzan, Jr., H. Computer Organization and the System/370. Von Nostrand Rheinhold, 1971.

48. Knuth, D. The Art of Computer Programming, Vol. 1 Fundamental Algorithms. Addison-Wesley, (2nd Printing) 1969.

49. Knuth, D. The Art of Computer Programming, Vol. 2, Seminumerical Algorithms. Addison-Wesley, 1969.

50. Knuth, D. The Art of Computer Programming. Vol. 3, Sorting and Searching. Addison-Wesley, In press.

51. Korfhage, R. Logic and Algorithms with Applications to the Computer and Information Sciences. John Wiley, 1966.

52. Laurie, E. J. Modern Computing Concepts - The IBM 360 Series. Southwestern, 1970.

53. Lefkovitz, D. File Structures for On-Line Systems. John Wiley, 1967.

54. Martin, J. Design of Real-Time Computer Systems. Prentice-Hall, 1967.

55. Martin, J. Telecommunications and the Computer. Prentice-Hall, 1969.

56. Martin, J. and Norman, A. R. D. The Computerized Society. Prentice-Hall, 1970.

57. Maurer, W. D. Programming: An Introduction to Computer Languages and Techniques. Holden-Day, 1968.

58. Meadow, C. The Analysis of Information Systems. John Wiley, 1967.

59. Minsky, M. Computation: Finite and Infinite Machines. Prentice-Hall, 1967.

60. Oettinger, A. G. and Marks, S. Run Computer Run. Harvard University Press, 1969.

61. Parkhill, D. The Challenge of the Computer Utility. Addison-Wesley, 1966.

62. Rice, J. K. and Rice, J. R. Introduction to Computer Science: Problems, Algorithms, Languages, Information and Computers. Holt, Rinehart and Winston, 1969.

63. Rosen, S. (Ed.) Programming Languages and Systems. McGraw-Hill, 1967.

64. Sammet, J. E. Programming Languages: History and Fundamentals. Prentice-Hall, 1969.

65. Sayers, A. P. (Ed.) Operating Systems Survey. Auerbach Corp., 1971.

66. Sprague, R. E. Information Utilities. Prentice-Hall, 1970.

67. Sterling, T. D. and Pollack, S. V. Computing and Computer Science. MacMillan, 1970.

68. Stimler, S. Real-Time Data-Processing Systems. McGraw-Hill, 1969.

69. Stone, H. S. Introduction to Computer Organization and Data Structures. McGraw-Hill, 1972.

70. Taviss, I. The Computer Impact. Prentice-Hall, 1970.

71. Teague, R. Computing Problems for FORTRAN Solution. Canfield Press, 1972.

72. Trakhtenbrot, B. A. Algorithms and Automatic Computing Machines. D. C. Heath, 1963.

73. Viavant, W. (Ed.) Readings in Computers and Society. Science Research Associates, In Press.

74. Walker, T. Introduction to Computer Science: An Interdisciplinary Approach. Allyn and Bacon, 1972.

75. Walker, T. and Cotterman, W. W. An Introduction to Computer Science and Algorithmic Processes. Allyn and Bacon, 1970.

76. Watson, R. W. Timesharing System Design Concepts. McGraw-Hill, 1970.

77. Wegner, P. Programming Languages, Information Structures and Machine Organization. McGraw-Hill, 1968.

78. Wilkes, M. V. Time-Sharing Computer Systems. American Elsevier, 1968.

79. In addition to the above list, several collections of articles originally appearing in "Scientific American" have been published in book form by W. H. Freeman and Company. Specifically, they are:

a) Information. 1966.

b) Computers and Computation. 1971.

80. Various conference proceedings, journals, bulletins, and the like should also be maintained in a library collection. The following are of special interest:

a) Communications of the ACM. (monthly)
b) Computing Reviews. ACM, Inc. (Monthly)
c) Computing Surveys. ACM, Inc. (Quarterly)
d) Proceedings, ACM National Conference. (Yearly)
e) Proceedings, Fall Joint Computer Conference. (Yearly)
f) Proceedings, Spring Joint Computer Conference. (Yearly)
g) Proceedings, IFIP Conference (Every three years)
h) Proceedings, IFIP World Conference on Computer Education, 1970.
i) Proceedings, Computers in Undergraduate Science Education.
j) SIGCSE Bulletin. (Available to members of ACM's Special Interest Group - Computer Science Education)
k) SIGCUE Bulletin. (Available to members of ACM's Special Interest Group - Computer Uses in Education)
l) SIGUCC Bulletin. (Available to members of ACM's Special Interest Group - University Computing Centers.)

ATTENDEES SIGUCC SYMPOSIUM

June 9-10, 1972
Sheraton Biltmore Hotel
Atlanta, Georgia

Bruce K. Alcorn
National Laboratory for Higher Education
Mutual Plaza
Durham, North Carolina 27701

Ronald Anton
Swarthmore College
Swarthmore, Pennsylvania 19081

Richard Austing
University of Maryland
Computer Science Center
College Park, Maryland 20742

J. P. Boulanger
PMC Colleges
Chester, Pennsylvania 19013

Dennis Brooks
Lamar Community College
Lamar, Colorado 81052

Harris Burns
Randolph-Macon College
Ashland, Virginia 23005

D. J. Clemens
Paducah Community College
Paducah, Kentucky

Edward Day
Fanshawe College
London, Ontario, Canada

Homer Easler
Aurora College
Aurora, Illinois 60507

Gerald Engel
Pennsylvania State University
University Park, Pennsylvania 16802

J. Engelhardt
Savannah State College
Savannah, Georgia 31406

H. James Farrell
P. O. Box 56
St. Mary's College
Winona, Minnesota 55987

Joseph B. Frechen
Department of Mathematics
St. John's University
Jamaica, New York 11432

Gerst Gibbon
Chatham College
Pittsburgh, Pennsylvania 15232

Harold Glander
Carroll College
Waukesha, Wisconsin 53186

John E. Haluska
St. Andrew's College
Laurinburg, North Carolina 28352

Preston Hammer
Pennsylvania State University
University Park, Pennsylvania 16802

Larry Henson
Box 2293
Berea College
Berea, Kentucky 40403

William Hightower
Armstrong State College
Savannah, Georgia 31406

Glenn Ingram
Washington State University
Pullman, Washington 99163

Thomas A. Jackson
Florida A&M
Tallahassee, Florida 32301

Whitney L. Johnson
State Council of Higher Education for Virginia
911 East Broad Street
Richmond, Virginia 23219

James L. Keefer
Elizabethtown College
Elizabethtown, Pennsylvania 17022

Tim Kelley
Southern Oregon College
Ashland, Oregon 97520

Wilhelm O. Kemp
Dickinson College
Carlisle, Pennsylvania 17013

Aaron Konstam
Linderwood Colleges
St. Charles, Missouri 63301

Robert Kyle
Emory University
Uppergate House
Atlanta, Georgia 30322

Alfred C. Lindem
University of North Dakota
Grand Forks, North Dakota 58201

Jacques LaFrance
Wheaton College
Wheaton, Illinois 60187

James McDonald
Jacobsen Computer Center
Morningside College
Sioux City, Iowa 51106

Albert Malveaux
Xavier University of Lousiana
New Orleans, Louisiana 70125

Sister Patricia Marshall
Xavier University of Louisiana
New Orleans, Louisana 70125

Jesse J. Mayes
1234 Massachusetts Avenue, N.W.
Federal City College
Washington, D. C. 20005

Pamela McGinley
Box 207
Nanuet, New York 10954

Charles Moulton
Beaver College
Glenside, Pennsylvania 19038

Linda Moulton
Beaver College
Glenside, Pennsylvania 19038

T. Robert Newlin
University of Missouri - K.C.
Kansas City, Missouri

Jack Nichols
Lamar Community College
Lamar, Colorado 81052

S. Ron Oliver
Wayne State College
Wayne, Nebraska 68787

Louis Parker
North Carolina Educational Computer Service
Research Triangle Park, North Carolina 27709

Charles W. Phillips
LeMoyne-Owen College
Memphis, Tennessee 38126

Marvin Philpott
Wittenberg University
Springfield, Ohio 45501

W. Taylor Reveley
Hampden-Sydney College
Hampden-Sydney, Virginia 23943

R. Waldo Roth
Taylor University
Upland, Indiana 46989

Gordon Sherman
Office of Computing Activities
National Science Foundation
Washington, D. C.

Charles R. Shomper
University of Iowa
Regional Computer Center
Iowa City, Iowa 52240

Robin W. Spock
Cuny
535 E. 80th Street
Board of Higher Education
New York, New York 10021

Donald F. Stanat
Kalamazoo College
Kalamazoo, Michigan

Paul W. Sutton
North Central College
Naperville, Illinois 60540

Daniel E. Todd, Jr.
Pembrooke State University
Pembrooke, North Carclina 28372

Robert A. Todd
Florida Keys Community College
Key West, Florida 33040

Richard A. Vogel
Western Maryland College
Westminster, Maryland 21157

Fred Weingarten
Pitzer College
The Claremont Colleges
Claremont, California 91711

James H. Westmoreland
University of Tennessee at Martin
Martin, Tennessee 38237

Richard Wood
Gettysburg College
Gettysburg, Pennsylvania 17325

Karl Zinn
U-M CRLT
109 East Madison Street
Ann Arbor, Michigan 48104