DOCUMENT RESUME

ED 031 936 24 EM 007 328

By-Ohlman, Herbert, Ed.

Proceedings of a Symposium on the Uses of Computers in Education.

Central Midwestern Regional Educational Lab., St. Ann, Mo.

Spons Agency-Office of Education (DHEW), Washington, D.C. Bureau of Research.

Bureau No-BR-6-2875

Pub Date 1 May 69

Contract-OEC-3-7-062875-3056

Note-136p.

EDRS Price MF-$0.75 HC-$6.90

Descriptors-Computational Linguistics, Computer Based Laboratories, Computers, Computer Science, *Computer Science Education, *Curriculum Development, Curriculum Evaluation, Curriculum Planning, Curriculum Research, Data Processing, Dial Access Information Systems, Electronic Data Processing, Information Processing, Input Output Devices, Machine Translation, *Time Sharing

Identifiers-*New England Educational Data Systems

If computers are to be used creatively in education, they should be made available to administrators, teachers, and students. Educational Data Processing (EDP) is a means to this end. Small and poor systems can join in cooperative county, state or regional computer systems efforts such as the New England Educational Data Systems (NEEDS) in whose philosophy EDP is a means to a more creative instructional use of computers, and whose financial structure is geared to this end. Alternatively, a school or system may "borrow," using a remote-access educational computing system: batch-processing or time-sharing computing. The Illinois Institute of Technology has developed courses in computer language and programing for high school students and teachers and has made its computer available to schools for $2300.00, including the cost of Bell Telephone equipment and installation. New York Institute of Technology's System for Individualizing and Optimizing Learning through Computer Management of the Educational Process synthesizes educationally related data such as media available resources, enabling teachers, students and administrators to choose and use wisely the best from available sources. Curriculum developers will have a scientific base for development. Multiprocessing and multiprograming concepts are also discussed. (MM)

TECHNICAL REPORT SERIES

NUMBER 6
1969

cemrel

CENTRAL MIDWESTERN REGIONAL EDUCATIONAL LABORATORY

Proceedings of a Symposium on the
USES OF COMPUTERS IN EDUCATION

edited by

HERBERT OHLMAN

FOR EDUCATIONAL

RESEARCH
INNOVATION
DIFFUSION
IMPLEMENTATION

Proceedings of a Symposium on the

USES OF COMPUTERS IN EDUCATION

Co-Sponsored by CEMREL and the State Departments of Education
Illinois, Kentucky, Missouri, Tennessee

February 13, 1968

edited by

Herbert Ohlman
Associate Director for Educational Information Systems
Central Midwestern Regional Educational Laboratory, Inc.

Speakers

Allen B. Ellis
NEEDS, Inc. and Harvard University

Alexander Schure
New York Institute of Technology

Peter Lykos
Illinois Institute of Technology

Herbert Bright
Computation Planning, Inc.

Chairman

Herbert Ohlman

Published by the Central Midwestern Regional Educational Laboratory, Inc., a private non-profit corporation supported in part as a regional educational laboratory by funds from the United States Office of Education, Department of Health, Education, and Welfare. The opinions expressed in this publication do not necessarily reflect the position or policy of the Office of Education, and no official endorsement by the Office of Education should be inferred.

Central Midwestern Regional Educational Laboratory, Inc.
10646 St. Charles Rock Road
St. Ann, Missouri 63074
314-429-3535

May, 1969

PREFACE

by

Herbert Ohlman

PREFACE

COMPUTERS IN EDUCATION: A YEAR LATER

A little over a year ago, CEMREL and the state departments of education of Illinois, Kentucky, Missouri, and Tennessee, organized a symposium on the educational uses of computers. Approximately 125 educators attended the meeting, which was held at the University City High School on February 13, 1968.

Our four invited speakers discussed a wide spectrum of applications: Dr. Ellis spoke on administrative and counseling uses; Dr. Schure on computer management of instruction; Dr. Lykos on student programming and problem-solving; and Mr. Bright gave a tutorial on third-generation computers.

We regret the delay in publishing these proceedings. The quality of the tape recordings made during the symposium was very poor, and the transcription of them into reasonably verbatim transcripts was a long and arduous task. Also, some speakers decided to revise their talks, so that only Dr. Ellis' and Dr. Lykos' papers remain essentially verbatim. However, we hope these changes have improved the readability and intelligibility of the material.

The arrangement of papers follows the sequence of the symposium. There is a section for each speaker, set off by a colored divider. Within each section is a speech, a discussion, and an appendix (where the speaker supplied additional material).

We hope that you will find the published proceedings helpful in planning how computer science and technology may best be utilized in your institution.

One result of the symposium was the starting of a project in computer literacy by CEMREL. Under our auspices, twenty Tennessee teachers were trained at the Illinois Institute of Technology during the past summer in an intensive workshop emphasizing IITRAN, a fast, low-cost, student-oriented programming language. Dr. Donald Thomsen and Dr. James Winter, CEMREL assistant directors responsible for program offices in Memphis and Nashville, respectively, had the operating responsibility for this project.

A number of these teachers have organized computer courses in their high schools. Because the greatest interest was shown in the Nashville area, a number of attempts were made by Dr. Winter to obtain access to a local computer which could support the IITRAN language. However, he finally had to install a Teletype in the Nashville CEMREL office and process the student programs via dial-up telephone on the IIT computer in Chicago.

Despite these difficulties, a number of schools have made significant progress in organizing courses in computer literacy. Details of this project will be made available in a forthcoming CEMREL document.

ACKNOWLEDGMENT

There are a host of people who helped me in organizing, holding, and publishing this symposium. Foremost are our four speakers, who covered a great deal of ground in a short time: Dr. Allan Ellis, Dr. Alexander Schure, Dr. Peter Lykos, and Mr. Herbert Bright.

The state departments of education for the four-state region which CEMREL covers were cosponsors of this symposium. They were represented by the heads of their data processing facilities: Mr. Charles Bratten from Kentucky, Mr. Donald Norwood from Illinois, Mr. Charles Pullen from Tennessee, and Mr. Jack Roy from Missouri. We appreciate their cooperation and interest in making this symposium a success.

Special thanks are due to Dr. Mark Boyer and the University City High School for their hospitality.

I also want to thank our sizeable and articulate audience for their participation. All attendees who signed our register are listed in Appendix I.

For CEMREL, I would like to acknowledge my debt to Dr. Wade Robinson, Dr. Donald Thomsen, and Mr. Andrew McCormick, who, with me, dreamed up the idea for this symposium and helped carry it out. Also, I'd like to acknowledge the help of Dr. Boyd Carter, Dr. Earl Morris, and Dr. James Winter of CEMREL's Program Offices, who saw to it that the symposium had both regional and national representation.

I'd also like to thank Mr. Glen McAlister and Mr. Dan Magidson for their help with facilities, and Mrs. Verna Smith and Mr. James Dean who shared with me the seemingly endless chores of editing.

Our willing staff of secretaries helped in numerous ways, especially Mrs. Charlotte Hayden, Mrs. Stephanie Macks, and Mrs. Leslie Osborn. In particular, I am grateful to my present secretary, Mrs. Susan Keck, for being so patient with the endless corrections and retyping which were essential to the successful publication of this symposium.

Herbert Ohlman
May 1, 1969

TABLE OF CONTENTS

WELCOME by Dr. Mark Boyer,
University City High School

and

GENERAL INTRODUCTION by
Herbert Ohlman

Welcome by Dr. Mark Boyer, Principal, University City Senior High School

Dr. Garrison, the superintendent of schools, has asked me to extend greetings to this group this morning, and it is indeed my pleasure to do so. It is our hope that your conference today will be a very profitable one. If there is anything that we can do to add to your comfort and convenience while you are here, please feel free to call upon us.

I would like to say just a few words about the school. Many of you are familiar with University City High School; we are a school of about 1,850 students. We have the unique distinction of being one of the high schools in the area that is becoming smaller instead of growing larger. We had an enrollment of 2,150 and will become as small as 1,800 before going back up to 1,900. It looks as though our enrollment will stabilize at this point.

We are endeavoring to move in the direction of individualized instruction. To achieve this goal we are attempting to build a more flexible organization in which teachers are involved in decision-making concerning class size and time, curriculum materials, teaching strategies, and other learning activities.

I hope that while you are here you might take some time to look around the building. It is an old high school. It was built in the late 20's but you will see, I think, some things that may be of interest to you. I particularly call to your attention some of the team rooms and the new library. The architect had the idea of covering a courtyard to provide space for a new library. I think it would be worth your time to see what can be done by covering over a courtyard and knocking down a few walls -- the result is a very attractive library.

The youngsters expect you to be here -- I told them this morning that we would have a large number of visitors. I asked them to be sure to help you if they were called upon to do so. Be sure and ask them some questions so that they won't feel that they are left out. Also, feel free to visit with some of the students about what they are doing. They will be very interested in talking to you.

Again, on behalf of the Board of Education, the administration, the staff, and the students, we welcome you to University City High School. Thank you.

General Introduction -- Herbert Ohlman

CEMREL is cosponsoring this Symposium with the four State Departments of Education in our region. They are represented here today by Mr. Charles Bratten from Kentucky, Mr. Charles Pullen from Tennessee, Mr. Donald Norwood from Illinois, and Mr. Jack Roy from Missouri.

A number of other regional educational labs are also represented. To my knowledge, there are people here from the Educational Development Center in Newton, Massachusetts; from the Center for Urban Education in New York City; from the Northwest Regional Educational Research Laboratory in Portland, Oregon; from the Mid-Continent Regional Educational Laboratory in Kansas City; from the Appalachia Educational Laboratory in Charleston, West Virginia and from the Southeastern Educational Laboratory in Atlanta. We also have guests from the U.S. Office of Education and from the National Science Foundation.

It is our particular good fortune to have four outstanding speakers on our program: Professor Allan Ellis of NEEDS and Harvard University; Dr. Alexander Schure, president of the New York Institute of Technology; Professor Peter Lykos of the Illinois Institute of Technology; and Mr. Herbert Bright, president of Computation Planning, Inc.

However, before we hear about their pioneering work in this field, I'd like to tell you something about CEMREL's interest and growing involvement in computer technology.

After existing for centuries without computers, why does education need these incredibly complex and costly tools? Among many reasons that could be put forth, five problem areas seem to me to merit our immediate consideration:

First of all, the increasing rate of growth of the population -- particularly that portion which is of school age, with their increasing longevity in educational institutions -- will likely outrun the supply of qualified teachers.

Second, recent trends towards individualizing instruction demand much more complex scheduling and materials management than we can cope with by hand.

Third, the complexity of the school and of the society in which it functions (and this complexity is aggravated by increasing school-district consolidation) demands new management and information tools at all levels if it is to work efficiently -- or perhaps if it is to work at all.

Fourth, the ever-growing mass of record keeping and reporting threatens to overwhelm the ability of clerks and administrators to deal with it.

Last -- but certainly not least -- the growing impact of the computer on society impels us to give our students a reasonably deep knowledge of what these intelligence amplifiers (as computers have been called) can and cannot do; perhaps even more vital to our survival is what they should and should not do.

CEMREL, working with the four State Departments of Education and with the U.S. Office of Education, intends to devote an increasing effort to bring the benefits of computer science and technology into this region. Today, we believe, marks the opening thrust of what we hope will become an area of unparalleled interest and utility to our schools and colleges.

Our four speakers represent a broad and a deep knowledge of educational computer uses. They will give us a new insight into this much misunderstood area, and I hope that we will leave the Symposium with our interests whetted to the point where we will seek out just how we can use computer science and technology in our varied positions and responsibilities creatively, knowledgeably, and wisely.

MULTISTATE EDUCATIONAL DATA PROCESSING

by

Allan B. Ellis

INTRODUCTION OF ALLAN ELLIS:

Herbert Ohlman: Our first speaker is Professor Allan Ellis. He is director of Area IV (Research and Instruction) for New England Education Data Systems (NEEDS), and also assistant professor of education at the Harvard Graduate School of Education, where he presently teaches two courses on computers in education. He is also a principal investigator and director of the computer area of the ISVD (Information Systems for Vocational Decision) project at Harvard.

Professor Ellis holds a doctorate in education from Harvard, and in his varied career has been a social studies teacher, a research assistant in statistics, a research assistant in programmed instruction, and a fellow of the Harvard Computation Center. He has published many papers on such topics as educational data banks and content analysis, and McGraw-Hill will soon publish his new book on computers in education.

Professor Ellis is membership chairman of the New England Chapter of the American Educational Research Association, and is educational representative from Harvard Graduate School of Business to the Association of Computing Machinery.

He is going to tell us about the potential benefits--and hazards--of developing and operating what is probably the first educational computer center to serve a multistate region.

MULTISTATE EDUCATIONAL DATA PROCESSING

Professor Ellis:

Grade reports, attendance registers, class schedules, test results, bus lists, health records, and the like come pell-mell from the school suggesting that the principal product of school is not as much the educated child as it is the printed document. These documents are all necessary, of course, if a school is to run efficiently and properly; but schools, after all, are for educating children, and it is little comfort that school personnel perform clerical tasks on behalf of students if in so doing their attention is diverted from these same students. Schoolmen know all too well that time must be found for processing school data and producing reports to students, state government agencies, school board members, parents, and all the other people who have some concern or other with what happens in schools. Recognizing, however. that this time cannot be borrowed from children -- there can be no paying back -- school administrators have begun to look to the computer for relief.

And, indeed, this relief has come. Computer programs exist to schedule schools, assign students to classes, calculate grades and print report cards, score and analyze tests, and, in general, perform many of the school's administrative and clerical tasks. This application of computers is called education data processing and abbreviated EDP.

Of all the applications of computers to educational problems, EDP is the least glamorous, appearing dull and ordinary when compared to other applications under development throughout the country. Researchers at the University of Connecticut, for example, are testing procedures for grading student essays by computer; at Harvard a group of professors is developing a computerized information system for vocational guidance; a University of Wisconsin researcher is exploring, by computer simulation, the mysteries of concept attainment in children; elementary school students in Palo Alto, California begin their day with computer controlled arithmetic drills. Exciting, new, promising, these uses of computers make the more routine education data processing, however advanced the technology, appear surprisingly conventional.

Yet, unglamorous as it may seem to use computers to figure grades and attendance or to print schedules and class lists, EDP obviously meets the immediate and considerable need felt by schools to remove burdensome clerical chores from staff members, whose time with students is too valuable to be shared by such matters. But, introducing EDP into schools has certain additional values which, while they may be hidden, are far more lasting; so much so that education data processing may be, in fact, the most important application of computers to schools.

One such hidden value of EDP concerns the mysteries, and thus the fears, which characterize common impressions about computers. Just as earlier men shied away from fire even as they deified it, school personnel and other modern men often mistrust and yet wonder at the powers of the computer, this latest "honor given to mortal men beyond their due." It is not surprising that listening to accounts of what computers are has done little to make the schoolman more familiar with computers or to leave him with any but an incomplete knowledge of what they can be made to do, for no amount of telling will do; it is through the day to day use of computers on familiar problems and in familiar settings that understanding will come. By seeing first-hand how computers can be used to solve problems he has grappled with, even though they be clerical and administrative, the schoolman will come to learn the lesson which others cannot teach him: computers are useful, not magical, and are, therefore, worthy of neither fear nor libation.

The knowledge gained from a day to day use of computers to automate clerical and administrative procedures is a beginning knowledge. As such it often leads to a second, more important value of EDP by affecting the schoolman's attitudes not about computers but about the educational procedures themselves. It does not take long for school personnel to imagine other, new applications of computers, as in the case of calculating grades and printing report cards where it soon becomes clear that computers can also produce frequency distributions of grades, or rank students by grade, or identify students whose grades are too low. These "new" applications need not be limited to clerical tasks, of course, but even if they are, they may lead in turn to notions of how the procedure itself might be altered. It may occur to a principal, for example, that the calculating ability of the computer makes it possible to compute student grades which are weighted by course level or difficulty. (Such grade point averaging, in fact, has become common among schools using EDP.) This idea may lead to other, more sophisticated rules for determining or reporting grades, conceivably to the point where grades or test scores are replaced by computer produced English sentences much the way Car Helm has described (1966). Thus, as schoolmen participate substantively in education data processing, they increasingly come to reconsider the very procedures that are being automated.

Beyond such hidden values, EDP, unlike other applications of computers to schools, is clean cut. The administrative and clerical procedures normally automated through EDP are, for the most part, well specified procedures. Most teachers could provide an accurate, even detailed, description of how they calculate a student's final grade from first quarter, second quarter, mid-term examination, third quarter, and final examination grades, and assistant principals or other central office personnel can set down unambiguously how they figure total days attendance, average daily membership, and the number of illegal days absence. Indeed, the reason these and similar procedures are distractions to schoolmen is that they are specified well enough for someone else to perform them.

With EDP, furthermore, schools can begin automation without making substantive changes to these procedures. Many of them need changing, of course, and and indeed EDP can be the medium for such change. Change comes gradually and and is not imposed from the outside, however, since new educational practices come as results of continued, knowledgeable use of computers. Thus, while EDP may not appear glamorous and exciting when compared to other applications of computers, it is nonetheless the appropriate beginning.

For most schoolmen, however, it does little good to realize that EDP is the way to begin. An average school system spends as much money a year educating thirty children as it would spend a month renting a typical computer. Even smaller, less powerful computers can cost as much as one hundred thousand dollars a year, not including space, air conditioning (required for proper machine functioning), and general overhead. Computers do not operate themselves, moreover; operators, programmers, keypunchers, coders, a supervisor, and, naturally, the involvement of school personnel are all necessary if a school is to have a data processing center. Recognizing EDP as the solution to one problem seems to do no more than pose yet another, equally perplexing one.

Large cities such as Boston, Chicago, Baltimore, and Cleveland do not have this difficulty. The number of school records that must be processed one way or another in these cities is sufficiently large to warrant the expense of even the most routine computer applications. Some other communities such as Shaker Heights, Ohio, or Palo Alto, California, while considerably smaller, are wealthy enough so that the cost of an EDP installation is not overly taxing on the school budget. These school systems are unusual, however, and for the typical superintendent other means must be found if his school's data processing is to be automated.

Some enterprising superintendents of schools have made arrangements with local insurance companies, supermarkets, banks, or other businesses to "borrow" their computer on off hours for school data processing. Other school systems purchase EDP services from such service bureaus as C.E.I.R., IBM's Service Bureau Corporation, and even McDonnell Aircraft Corporation in St. Louis.

Another possible solution to the problem of establishing EDP in the schools without incurring excessive costs, and which is becoming increasingly widespread, is the joint establishment of an EDP center to serve more than one school system. A recently published book -- already outdated -- by Goodlad, O'Toole, and Tyler* lists several examples of county and state-wide EDP centers formed cooperatively to provide data processing functions for a number of systems. Prominent among these are the Iowa Educational Information Center, and the state-wide data processing networks set up by departments of education in California, Connecticut, New York, and Hawaii.

Of all the cooperatively-formed EDP centers in the United States, one is regional. This center, called NEEDS, which stands for New England Education Data Systems, was established in 1960 to serve school systems in Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, and Vermont. Two large grants from the Ford Foundation -- one in 1963, another in 1966 -- have made it possible for NEEDS to automate record keeping for more than one hundred thousand students in New England while keeping the costs to member schools at the substantially reduced price of $3.50 a year for each student.

Among cooperative EDP centers, NEEDS, as a regional center, runs the biggest risk of failure. This danger does not concern the center's ability to func-

* "Computers and Information Systems in Education," Harcourt, Brace & World, Inc., 1966.

tion and to bring EDP to schools. On the contrary, NEEDS, as other centers, does more than any local installation could do to develop such applications. Centralizing permits the hiring of highly qualified people, the development of more sophisticated computer systems, and, in general, a wider perspective on the problems attending the application of computers to schools. But, these are merely the outward signs of EDP, meeting a particular need of schools, to be sure, yet doing little in themselves to extend the influence of these applications. Going beyond mere application requires that schools play the major part, internalizing EDP so that its potential benefits and hidden values can be realized. Thus the success or failure of NEEDS, or any regional center for EDP, depends most heavily upon whether it is a means or an end.

Centers for which EDP is the end degenerate quickly into service bureaus, offering well itemized services to which individual prices are attached. The school as the customer may always be right, but it never steps behind the counter because a service bureau is a business. It may not be a profit-making business, but it has proprieties -- secrets which it may not consciously keep but which it goes to no trouble to divulge. Even a data processing facility within a school system can be a service bureau if it "provides" school personnel with EDP and does little to clarify for them the role computers can play in schools. When EDP is an end in itself, schools forfeit their equity, and what they receive amounts to outside help for which they pay the price of continued dependence.

NEEDS can be distinguished from other data processing centers not by the fact that it was the first, or that it is regional, or even that its programs are in the public domain. Most important of all about NEEDS is that it is wholeheartedly a means and not an end. NEEDS extends the notion that EDP leads to a better understanding of how computers can be applied to education by saying that EDP is the first step toward creating "the machinery by which the research of universities may effectively be related to the ordinary life of school systems."* To be sure, NEEDS exists to meet a present need; but, to them, EDP is very much a beginning -- an excuse to get to other things. This goal requires a different relationship between NEEDS and the school systems and, of course, has a considerable effect upon its organization and its functioning.

One effect this goal has upon the operation of NEEDS is so complete that it is almost correct to say when a school system joins NEEDS it agrees to provide data processing services to itself. This seemingly peculiar circumstance stems from the fact that joining NEEDS means agreeing to participate in NEEDS and not merely to accept services from it. NEED's point to this, of course, is that by maintaining an active responsibility for how computers are applied to their schools, schoolmen forfeit no equity. Far from being customers, schools become partners, participating in all aspects of the venture.

At the most detailed, day to day level of school participation in NEEDS is the coordinator. One professional staff member from each of the sixty-four New England school systems comprising NEED's membership is charged by his superintendent with the responsibility to oversee and coordinate data processing for that system. Working with NEEDS personnel, the coordinator decides --

* "Report on the New England Education Data Systems," NEEDS, Cambridge, Mass., Nov. 1966.

often with counsel from other members of his school system's staff -- all matters of form, substance, and change. When, for example, a computer run produces a list of schedule conflicts -- students who could not be assigned to classes because of some difficulty either with the school schedule or with the students' course requests -- the coordinator examines the "output" and determines what changes are to be made to accommodate these students. While he works closely with NEEDS personnel, it is he and not they who decide. You might say, the coordinator is a NEEDS staff member although he functions more like an ambassador in his embassy who, while situated within the boundaries of a foreign country, is yet on his own soil.

School participation exists at other levels as well. Several superintendents of schools, for example, sit on the NEEDS Board of Directors and thus have a strong voice in determining policy. These superintendents share this responsibility with university professors, commissioners of education, and other leaders of education in New England all of whom comprise the directorate of NEEDS.

Between the detailed involvement of the school coordinator and the school's representation on the NEEDS Board of Directors, there exists another aspect of school participation in the form of the NEEDS coordinating committees. There are three such coordinating committees at NEEDS representing each of its major interests: schools, universities, and state departments of education. One function of the school coordinating committee -- whose membership varies from principals and other administrators to guidance personnel and teachers -- is to evaluate present operations and to consider possibilities to enhance them. A second of its functions is to devise meaningful ways to extend computer applications to other areas of the school. The coordinating committee, thereby, links the school's involvement at the policy level and at the level of day to day operations.

This extensive participating of school personnel is not the only way the goal of enhancing the schoolman's view of computers affects NEEDS' operations. Naturally, to participate wisely he must know what he is doing and to this end NEEDS spends considerable effort in what it calls "in-service training".

Formal institutes for coordinators begin with a comprehensive introduction to computers, covering such topics as the NEEDS operating system, use of unit-record equipment, computer programming, and the role of the computer in education. Such institutes, which are given regularly, are augmented by private consultation with individual coordinators where issues relevant to his school system are discussed.

NEEDS does not direct such educational activities to coordinators alone. Superintendents, guidance personnel, principals and their assistants, central office staff, and teachers all have enrolled in specially formulated institutes conducted at NEEDS. For administrators the topics are more general, designed to give them an overall view of automation in education, while for guidance counselors and teachers the topics are directed toward their particular interests: automating cumulative records, test scoring and analysis, grade reporting, and the like.

These educational activities extend beyond NEEDS' premises to such institutions as Harvard University and Boston University where several NEEDS staff members hold academic appointments. These individuals teach formal courses in education data processing and in educational research which are attended not only by university students, but by many NEEDS members as well. In this way NEEDS provides instruction in a wide range of topics relating to the uses of computers in education, and, thus, goes out of its way to keep no secrets and to conjure no mysteries.

These two features -- the participation and the education of school personnel -- establish that NEEDS is not proprietary and therefore not self-perpetuating as are service bureaus. Of course, a service bureau may well serve the data processing needs of a school; but NEEDS, while accomplishing as much makes no effort to spare schools the troubles and the burdens of responsibility. The education of school personnel and their extensive participation are steps toward the point where schools will possess the know-how to begin to do the job themselves.

In this spirit, NEEDS member schools are performing more and more of the routine data processing tasks on their own premises. The creation and maintenance of files, the pre-processing of pupil course requests, test answer cards, and master schedules, even the listing of teacher and student schedules are performed by some member school system at their own data processing centers, leaving NEEDS only the more sophisticated computer processing. In one case -- that of the Manchester, Connecticut Public School System -- NEEDS has established a sub-regional center whereby those schools in the vicinity can be serviced directly by Manchester personnel and need not bring school data to the main facilities in Cambridge, Massachusetts. The Manchester Data Processing Center is not a branch of NEEDS; however, it is a further step in a local school system's evolution to self-sufficiency in EDP. Thus regionalization becomes not a force for centralization, but rather a device for developing local capability. Far from establishing an EDP "network" under its central control, then, NEEDS to succeed as a regional center for EDP must relinquish such control and, in effect, be dedicated to going out of the EDP business.

But other schools will want to join NEEDS, expecting, as new members, to go through the process from the beginning. Old members, furthermore, will have developed enough to look toward extending EDP applications in their schools. For these schools, then, whatever their experience with EDP, continued access to the talents and facilities of a place like NEEDS is vital. So going out of the EDP business cannot mean closing shop since NEEDS' activities can be predicted to grow indefinitely. Instead, going out of the EDP business means reaching the point where EDP becomes second to the more important work for which it is an excuse.

This work, for NEEDS, is research and is based on the fact that while the product of EDP is the printed document, its byproduct is data. A report card printing run leaves behind on computer tape such things as course name and number, meeting times, rooms, teachers' names, students' grades, credits, previous credits, and attendance figures. Other EDP applications provide teacher and room utilization figures, student schedules, responses to test items, class standings, and the like. Such data, far from being merely the slag and dross of EDP, are the elements from which research can begin to be fashioned.

Of the two major aspects of research at NEEDS, the first is data analysis. Frequency distributions of grades either by class, or teacher, or course, test item analysis, summaries of facilities and personnel utilization, classifications of failures, and examinations of attendance patterns are some examples of data analysis performed at NEEDS. Beyond these simple analyses which NEEDS performs on byproduct data, there are others which require the collection of additional data. For example, NEEDS, in conjunction with a local school system, is examining how well a statistical model compares to present, more intuitive methods of assigning students to curriculum levels within courses. To do this, it was necessary to collect data on existing criteria, past performance of students, and the like. But, whether byproduct data are used exclusively or are joined by data specially collected, EDP remains the impetus for data analysis.

The second aspect of research at NEEDS centers on the development of new applications of computers to education. Within this area, the major activity of NEEDS' division of research is the creation of a computer based information system for vocational decision.* This new application of computers to guidance -- being explored in conjunction with Harvard University's Graduate School of Education -- will give students access to a large storehouse of data relevant to career choice. This access to computer-stored routines will be by way of consoles connected to a central computer through telephone lines, thus permitting schools to share in the project without needing to bear the major burden of cost. An outgrowth of this project will be a comprehensive set of computer programs useful not only for guidance, but for instruction and administrative decision-making as well.

At NEEDS, then, what starts out as the day to day processing of school data becomes the exploration of new applications of computers in schools. Thus EDP need not provide schoolmen with relief and nothing more. On the contrary, while EDP may not number among the glamorous applications of computers to education, it can be the starting point and indeed, the impetus for such applications.

---

* "The Computer and Career Decisions," by A. B. Ellis and C. S. Wetherell, Technical Memorandum 1, Harvard University Information System for Vocational Decisions, 1966.

DISCUSSION: ALLAN ELLIS

Question: This may sound naive and bread and butter oriented, but it seems quite obvious to me that you have to attain a certain critical mass before you can get this whole notion off the ground, in terms of funding. You've given us an idea what happens after the critical mass accumulation resumes, but not all areas of the country are going to be as far along in a lot of ways as New England. Have you any words to the wise who might be interested in launching such an effort?

Reply: My understanding of the question in that what I have given is a reasonable statement of how a regional center can function, but it's after the fact, in that other parts of the country -- as they begin to develop such centers -- will naturally face a lot of problems, and would I offer some words to the wise?

We almost didn't make it for a lot of reasons. We almost didn't make it philosophically, because no matter how glorious your goals are when you start, you have to be concerned with doing the job that you're doing, and doing it right, and doing it on time. Which means that you have to look like a service bureau, and you have to care about getting the report cards on time where they are supposed to be, under all circumstances -- and it is very easy to get caught up in that activity.

I don't see any major problem of getting started, of having schools get access to people through the regional laboratory. For example, people who have had experience developing such centers. Many of the computer programs that you would need exist already, and are in the public domain. Perhaps I am begging the question, but my answer would be that the only real danger is that you lose sight of the more glorious goal and end up being a service bureau -- and I say that having just lived through it. Well, we started in 1960; I would say that it was not until 1967 that we could honestly say we had gone beyond the service bureau stage and were beginning to look into a larger context. I would assume that it would take you less time than that -- perhaps two years.

Question: At present, must all data be brought into the bureau to be processed? That is, are there no remote terminals?

Reply: Yes, no remote terminals.

Question: Are there in the future to be remote terminals in the schools, or is this a problem?

Reply: It is a problem, but it's a problem we have tended to postpone. I am not sure whether we have postponed it or ignored it. If you were to say that you were interested in developing an educational data processing center to do clerical tasks for the schools, NEEDS would not be the first place I would suggest that you go to find out about doing it. And the reason for this stems from the fact that education data processing is for us an excuse. Since it is an excuse, we are not really overanxious to develop a widespread, in-depth set of applications in data processing. We feel that as soon as we can begin to give schools the local capability and to have them develop independently that the time that we would normally spend on new data processing applications (maybe in the financial area of the inventory area) could better be spent in exploring applications to the educational program. For example, we made a choice in taking on this extensive guidance project -- between it and a project that would extend our activities in data processing. We chose the one in guidance, feeling that schools can begin to provide the services to themselves, as Manchester has shown. Therefore, we think in terms of subregional centers like Manchester, where maybe ten school systems that are very close together share the burdens and the responsibilities, and thus we're able to postpone the question of data transmission.

We are, however, quite interested in the use of remote terminals for other than data processing activities. I see a great resemblance between the problems that we are trying to solve now in developing a computer-based system for career decisions that will permit students to sit at remote terminals and interact with data on computers -- and problems that people interested in computer-aided instruction face; and also to problems that people interested in administrative uses of terminals face. And so while we are postponing the issue of data transmission, we are quite interested in looking at possibilities of school administrators making use of remote terminals for a more dynamic access to their data.

Question: Could you explain in more detail the vocational guidance retrieval project?

Reply:* The USOE and Harvard University entered into a contract to have Harvard develop a prototype of a computer-based information system for vocational decisions, and this project is called ISVD, for short. Its director is Professor David Tiedeman, professor of guidance at the Graduate School of Education, and there are a number of other principal investigators in the project. There are a lot of ways to describe this project, and the simplest way is to say that it consists of two things, a student and a set of data, and the problem is to get the student access to the data, and the data access to the student.

Professor Tiedeman is interested in developing an environment where students can make decisions about what they are going to do next semester, or whether or not they are going to college or what they want to be (what professions or occupations they want to pursue). And he is interested not only in giving them access to the data, but also in developing their familiarity with the decision-making process.

Now the reason that I prefaced that, is that the desire to do both things is the thing that gets us into most of our trouble. The problem is not merely collecting data on occupations and relevant areas and letting kids browse

* Also see the appendix: "An Introduction to ISVD," following this discussion.

through them, but it is also an additional problem to provide the student access to the data through sets of routines that carefully and systematically reflect Tiedeman's notion of career decision-making. To do this, we chose to use a computer as a device between the student and the data, and we have established careful arguments of why we use the computer instead of something else, like a human being for example.

The data consists of roughly five categories of things. We have an occupational data base of 1,000 occupations, and something like 45 categories of information about each occupation. Another data base we have -- we call it the military data base -- consists of similar facts about occupations in the Navy, Army, and the Air Force.

We also have a number of data bases that we classify as student data bases. We have follow-up data on students who were graduated from the Newton School System within the last ten years. We have data on students who are now in the Newton School System, and we have data on those students that make up our population of users who are also from the Newton System. We also have information on colleges and universities, and junior colleges, and trade schools describing things as: Who goes there? How do you get in? What things can you concentrate on if you go there? And we are beginning to develop expectancy tables based on the experiences of the Newton graduate: What proportion of students who look like you, succeed to this extent or to that extent?

In order to have the guidance personnel determine how the student would get access to data, and how these data would be presented to him it was necessary to get some statements from them, unambiguous statements on how they wanted this machine to behave, and I found it difficult to get unambiguous statements from guidance people I spoke with because they weren't used to talking the way computer people think other people should talk, and so in order to provide them with a way of telling unambiguously what they want to do, we invented a pencil and paper language which we call "Minorca," which made it possible for them to specify precisely how they wanted the computer to behave, and yet to specify in language that is familiar to guidance people, and uncomputer-like. It was so successful that we have decided to computerize the language, and that is one of the activities we are engaged in in the project, and one of the bases on which I argued before -- that what we are doing for guidance has relevance for instruction, since this language behaves precisely like a computer-assisted instruction author language, except that it has certain differences and in my opinion is more powerful than many existing author languages.

Question: Are there other organizations such as these in other parts of the country? Do you know the men who established other such organizations? Do you think that perhaps there is an answer in making greater use of the computer in educational establishments other than organizations such as NEEDS?

Reply: I don't know in what order I will answer your questions. I'll say first: Yes, I think that with certain qualifications, one answer to getting computers applied in a more and more widespread manner to school systems is the establishment of other NEEDS throughout the country. I say this with certain qualifications, because, first, I am not entirely sure that all of the philosophy of NEEDS is pertinent to other parts of the country.

Yes, there are other centers throughout the country. The Iowa Educational Information Center, for example, behaves as though it were a regional center, even though it is a state-wide center; they have many national ties. In fact, the Iowa Center comes to mind when I say that all of NEEDS' philosophy might not be pertinent. One of our major interests was to establish more effective communication among people involved in New England at the state department of education level, and the school level, and the university level. Well, in Iowa that would be a waste of time since everybody at those levels talks to each other very well. In fact, the Iowa Center, just as MRC (Measurements Research Center) and other such educational institutions in Iowa, are governed by a governing board of university people, state department people, and the chief state school officer.

Also, in Canada, there are a number of such regional centers. I would suggest (even though it is outdated) that you look at the reference I made to the book by John Goodlad and others, called "Computers and Information Systems in Education."

Your middle question was: Are there now efforts going on to develop such centers? I will make passing reference to two that I know of, and perhaps let Al Coulson, if he's here, say a little more about one of them.

Of course, the state of California has for some time been working on the development of what they call regional centers throughout the state, and have been quite vigorous and done extensive work in the problems of developing new centers and coordinating them. Al Coulson, who is a member of Instructional Systems Division of RCA, located in Palo Alto, California, was telling me about some of their efforts in this area. Al, could you say a few words?

A. Coulson: The state of Texas has moved in this direction and is developing twenty service regions, and they are initiating data processing and some other uses of computing in the school systems, and that largely is being done with state support and ESEA Title III funds.

A. Ellis: I'd say without hesitation that CEMREL has all the contacts to the people in the country that you would ever need if you were interested in developing such centers, or even modifying them to fit your purposes better -- but still to build on work that others have done. I hope that answers your questions.

H. Ohlman: I just want to say a word about CEMREL, and what I believe is the general outlook of the regional education labs as far as centralization is concerned. You made reference to us previously as perhaps central-oriented. I don't really think that this is the whole story, since our main objective is to develop an optimum mixture of centralized service and decentralized access, and just what form this may take remains to be seen. Do you have any comments, Dr. Schure?

A. Schure: I think it's worth a remark about the work of the New York State Education Department. The New York State Education Department is developing a very comprehensive data-reporting network and many of the local school districts in Nassau County are to become stations in this integrated data network.

A. Ellis: Yes, I think I've given the State Departments short shrift. I apologize for that. I think that I mentioned only briefly that a number of state departments are very active in this area. Certainly New York, Iowa, and California did a considerable amount of work -- and many others. I always get myself in trouble when I try to list them because I leave out more than I include.

Question: What kind of trouble do we have allowing flexibility of forms and reporting among our membership; and in such a large area, with so many people, how do we handle trouble with peak loads?

Reply: There are a number of respects in which we are susceptible to trouble from this flexibility problem. The first and most obvious, and the very first one we had in fact, was that I've discovered that school superintendents are not the best people in the world to try to tell what to do, especially when what you're trying to tell them to do, they know from experience is not as good as what they are doing!

Fortunately, we sensed that that would be a problem, and therefore, decided to take the burden upon ourselves to generate computer programs that would permit all the flexibility we need. Now, if any of you know anything about computer programming, you know that that's probably an insane point of view; it's certainly an expensive point of view. In fact, a large factor in excessive costs that we had in the first couple of years of our operation came from the fact that we were writing computer programs and producing forms for report cards, for example, that would permit schools to have eight marking periods, or two; or to grade in letters or numbers, or a combination; to have remarks or to not have remarks; to have conduct indicators or not -- and they could be numbers or characters or abbreviations -- and so on. Not only is it a problem in just printing it out differently, but you have to do different calculations, and so on.

We have had similar problems with all of our systems. Our argument ran as follows: We are a membership organization, and our function is not at all to have the schools compromise in what they're doing, but to try to enhance their ability to do what they want to do, and therefore, the cost is justified. After a lot of soul searching, we then produced the computer programs that permitted this. It's an administrative or operational problem, too, because you have to have different card setups for different schools, and so on. We have lived through it; we feel that if we had to do it over again we would, and that it's much better than trying to have the schools conform to a single format and a single letter-grading procedure. I think that from a political point of view, it's absolutely necessary that even if you end up trying to get schools to change what they're doing, in the beginning you have to take schools on their own terms -- because people know what they are doing, and they've been doing it for some time, and it's just unreasonable to try to have them change by imposing change upon them. Yet, we feel that there are a lot of questions that could be raised about the present procedures, and our only alternative is to take the schools on whatever terms they're willing to come, and hope through becoming familiar with the capabilities of computers that these changes will begin more and more to occur to them, and that we can work together in creating changes.

I don't know what kind of answer that is to your first question, but another problem we have in flexibility is one where schools say: We know that you offer these different services -- all we want is to get our school scheduled by you, and we'll do our report cards ourselves. And some schools say: Do that and charge us less money! And we say: No! We can't do that, because the money we charge is not based on our operating costs, it's based on a realistic notion of what schools are willing to pay, and that the Ford Foundation is helping us, and that the operation costs are much greater than $3.50 a student.

Some school systems say: We don't want report cards automated, and we don't want our school schedule -- all we want is automated attendance -- but we'll pay for everything. And our answer to that is also: No! Because the minute we start to do that, we become surely for all time a service bureau, because you need the concept of membership in an organization that's attempting to explore change in schools, and you need as much commitment and activity on the part of the schoolmen before you can hope to have any change. So that kind of call for flexibility has also gotten us into potential trouble.

Now the second question is about peak loads. We have peak loads; report card time, every six weeks -- considerable peak, and during the summer -- particularly the end of the summer -- there is considerable peak on scheduling and getting class lists out, and so on. We have made some efforts to lower the peaks by distributing the work around the year, but there are certain reasons which I think at first glance make that unreasonable. One is that it begins to ask the schools to change their schedules, and they may not be too interested in that; second, during the slack periods, there are a lot of things that we can do. That is, we now have ups and downs, particularly because we're not doing as much as we should be doing. For example, we could take advantage of the low periods by introducing activities to elementary-school students, or by using our computer to explore some existing computer-aided instruction demonstrations that exist, to give schools a chance to look at some other applications.

I am not giving you the answer of a data processing manager -- because if I was a data processing manager I'd give you a straight answer. We, however, survive the peaks. Up to this year we had a 1401 with 12,000 characters of storage, and two 1311 disk-pack drives, which means offline storage of about 5 million characters, and it was a strain, and we had to spend a considerable amount of our effort on outside computers -- like Harvard University's 7094, which is a very large machine. Since then, however, we have gotten an RCA Spectra 70/45 computer, which is a respectable intermediate size computer -- considerably more powerful than a 1401. Right now we are in the very expensive, very time-consuming stage of reprogramming ourselves, which is a problem. But, we see that much of the difficulties will be alleviated -- the peak problems will be alleviated -- with this machine. Right now, we are using this machine to emulate a 1401. In the computer business, it's possible to get one machine to look like another one, and behave much like another one, and the Spectra 70/45 is capable of imitating two 1401's, so even with that we've reduced our burden of the flow of data considerably.

Question: A minute ago, you made the statement that you're given the data bank, and that you're in the process of creating expectancy tables that students may use. Now, algorithmically, the data is compiled, and is made available to students at consoles. What is being done to help the student recognize that he will be able to use the data?

Reply: The question is: What is being done in this guidance project to help the student to become better able to use all the data that we're giving him access to.

What I meant earlier when I said that Tiedeman is not just interested in having data for the students to get at, but he's interested as well in creating a decision-making structure, which would be the filter through which they access the data. I was, I guess -- cryptically saying -- that our major concern is, in fact, to get students to not only learn how to make decisions better -- going through the act of it better, and to learn about ourselves more -- but also to develop what Tiedeman calls a sense of agency, which is a notion that you as an individual are an agent in what happens to you. So to be more specific, guidance people wrote what we called scripts -- sets of statements of what will happen, what kind of interaction the student will have with the machine -- and some of these scripts behave very much like programmed instructional sequences, in that they discuss them with him in a purely linear way, or in a purely specified way. That is, the script that the student gets is a function of the questions he is asking, and the stage that he is at in his decision-making process.

Furthermore, we have developed a semester course of the computer -- I think it will end up being a year-long course -- for junior high school students at Newton, and we've taught it for one term so far, on career decision-making, covering what steps are involved; what sorts of questions people ought to ask themselves or tend to ask themselves; what probabilities are and what they mean and how you should treat them; when you take them seriously, and when you don't; the notion of error in probabilities; and so on. And I think one very important idea is the following: that we try and get students to understand and appreciate that data in the real world are never complete, and in fact, it i probably this incompleteness of data that accounts for the needs for decisions most of the time. The minute you accept the fact that data are not complete, and you accept the fact that you have to make a decision, if you realize that this decision is not being imposed on you by the outside -- because it is not determined by the outside -- we hope the student will begin to feel that he is the one who is responsible for the decision that's made, meaning he has to take the consequences, or the credit, depending on how things work out, and that he is an agent in what is happening to him, and that therefore he is free -- in one important sense of the word.

I have given you a very very inadequate statement of our efforts to have the student examine -- to learn about the process, and to examine himself -- and I apologize for that.

One final thing I would say is that: One of the important ideas of our project is to develop within the computer the capability of monitoring the student's performance. Meaning by this -- not only that we keep track of what the student is doing, and what answers he gets, and what questions he asks -- but also that we give him what we know about him; that is, we give him the monitored data to process, so that he not only makes decisions, but he also begins to become aware of himself making decisions, and uses things that we can hopefully provide him about his style of decision-making, or his propensities in decision-making, that he might use as further data in coming to decisions.

Question: You mentioned in connection with this guidance program that your group is interested in the relative effectiveness of presenting these areas of guidance materials -- relative effectiveness compared to other means of guidance, of which the computer is being compared to others. Do you have any comment on any of this?

Reply: No, I don't. The comment that I made should have been made more carefully. We began by saying that we would like to create a reckoning environment in which we can put a student. This reckoning environment being such that it displays by its functioning, characteristics of reckoning or deciding, that because large amounts of data are involved requiring processing, sorting, juxtaposing large amounts of data, keeping track of many students at once, and so on, that it's quite likely that a human being is not an appropriate master element in this reckoning environment, and that, therefore, we need, ideally, a guidance machine, or a reckoning machine.

We conclude that there is no such machine, but we happen to know something interesting about computers, even though a computer is to some extent a poor excuse for a potential reckoning machine. The most interesting thing that I can think of about computers is that when you buy them and plug them in the wall, they don't work! And the reason that they don't work is of profound importance, because, in fact, computers are not machines at all! What they really are, are devices that read descriptions of machines -- we call them computer programs and think of them as instructions of how computers are to behave -- but, in fact, you can think of a computer program as a blueprint of a machine, and what computers do is they read your description of a machine and they imitate that machine -- they become that machine. So if I write a computer program to print report cards, another way I can think of it is that I am writing a procedure for printing report cards, but this not only defines the procedure, but the computer then builds itself into a report-card-printing machine, and prints out the stuff, and that operationally there's no difference between a report-card-printing machine that I might build for real over here with pliers and screwdrivers, and a computer behaving like one. So that we say that we need to have a guidance machine, or a reckoning machine in this environment.

We're not entirely sure what this machine should do, because we don't know enough yet about the business, but we have some notions, and so our effort is to turn a computer into a guidance machine. And we do it by creating a certain environment in the machine that will make it capable of reading these descriptions the way we want them read, and to be able to operate, effectively.

Thus, we have a computer that's able to read the scripts that I mentioned before, which are descriptions -- not only descriptions of how the student is to be given access to the data -- but it's, in fact, a description of how a guidance machine should behave. I apologize for that long-winded statement, but that's one possible answer to your question anyway.

A. Schure: Sometimes when you buy a computing machine and set it up, and you program it to do something, sometimes it doesn't work -- yet there may be many hidden initial manpower, capital, and other costs -- so it can't very well be off the line. It is a function of the human, and an extension of the human, and in that sense it's long term success is governed by the nature of the cleverness of what is programmed into it. This is true of all kinds of computing.

APPENDIX: AN INTRODUCTION TO ISVD

The Information System for Vocational Decisions, which we call ISVD, began the first of June, 1966 as the result of an agreement between the United States Office of Education and the President and Fellows of Harvard College. This agreement states that the members of the ISVD Project will construct a prototype of a computer-based system for use primarily in junior high school and senior high school guidance departments in New England. This prototype is to be delivered on or before the first of July, 1969, which is three years and one month from the day the ISVD Project began.

Cooperating in the ISVD Project, along with Harvard University, are the New England Education Data Systems and the Newton School Department. Furthermore it is being conducted in liaison with the Division of Vocational Education of the Department of Education of the Commonwealth of Massachusetts as represented by Walter Markham, the Director, and John Morine, the Senior Supervisor of Occupational Information and Vocational Guidance. The major objective of ISVD is to improve vocational decision-making through the use of a computer-based training system. The program is to be so designed that the student can relate knowledge about himself to data about education, training, and work and thereby create a body of information on which he can base his career decision. The program will link student, computer, and counselor in such a way that the student can conduct a dialogue with the computer while the counselor assists in interpreting and evaluating the results of that dialogue.

The original proposal for the establishment of ISVD which was submitted to the United States Office of Education poses the problem in the following way:

". . . participation in an occupation involves more than training in the specific skills required. Before, during, and even after vocational training the process of decision-making must also be involved. Central in decisions about occupations, jobs, or courses of study are facts about one's self and about work. Facts illumine and frequently create decisions. But decisions also create facts. This project therefore provides opportunity to study the interaction of facts and decisions, and their subsequent creation of information.

"The proposed ISVD will deliberately play upon a potentially useful distinction between data and information. The task of the information system is to enable the individual to transform data into information. This is to be done by teaching him to interpret the data in the light of his own knowledge, experience, and intention, so that his organization and use of the data represents his own personal relationship to them in the process of decision-making. We presume that only when data are used in this way can they be described as information where the individual is concerned. The information so generated can then, in turn, serve as data in the making of future decisions.

"Given that the quality of decisions is directly related to the kind, quality, and comprehensiveness of the data considered by the individual during the process of decision-making, then a fundamental task of guidance is to identify, evaluate, and classify needed data and to make them readily available to counselors and students in usable forms and at needed times and places. A second task is to learn how past decisions can be used to create information of value to the students who have made those decisions.

"Guidance workers have had difficulty in providing and effectively displaying data. This is so because the amount of these data is directly related to the unparalleled rate of change in the technological world, which in turn is rapidly producing basic changes in our society. If we are to prepare students with skills, and attitudes and understandings for a changed and continuously changing future, we must know something of the nature of the changes involved. We must also encourage students to think of vocational planning as a lifetime process, not a one-time decision. 'The counselor must think future and not experience or he will be of diminishing value to the student of the sixties and seventies' (Wrenn, 1962, p.20).

"Not only have counselors found it difficult to provide and display data, but the relatively infrequent contact between student and counselor has made the student's interpretation of data largely a hit-or-miss affair. Most students in secondary schools see a counselor three or four times a year at most. Furthermore, the nature of these contacts is frequently governed by a concern for the immediate next decision to be made and the immediate interest of the student. The amount and quality of facts available to the counselor at the time of an interview is limited by his own knowledge and his school's resources. And the counselor's usual function is to provide facts for the student at the same time (and frequently without distinction) that he is attempting to get the student to use them.

"What is needed is a system which will provide for the student direct access to all relevant facts without requiring the direct mediation of a counselor. This would bring about a change in the counselor's role. Instead of being both source and interpreter of facts, he would have the primary responsibility of interpreting the student's use of the facts as he transforms them into information. This would require attention to the role of unconscious motivation, and the effort to help the student transform his tacit understandings into explicit ones. Also included in his role would be training the student in the use of the data system, supervising him in it· use, and evaluating the student's decision-making process. Ultimately, it should be possible for the student to use the data system in a relatively independent manner for both exploration and decision-making, with recourse to the counselor when assistance or interpretation is needed."

The purpose of this project is to develop an information system for career choice built on a paradigm of decision-making proposed by David Tiedeman and Robert O'Hara. This paradigm depicts the behavior of an individual from his earliest confrontation with a choice situation, to his final analysis of the results. In its simplest form, it states that there are two major components to decision-making--exploration and choice.

An individual, having recognized that there is a problem he must solve, enters the stage of exploration--sizing up alternatives and options; and locating risks and consequences--with the aim of structuring and ordering the universe of choices. The individual asks: What is available? What are the possibilities? How do they relate to my goals? What are my goals? How do the alternatives relate to each other? What is the universe from which I can reasonably select a course of action? Once these questions start to be answered, the individual is ready to consider choice.

One element in the stage of choice is the moment of choice--that instant at which commitment is made and begins to grow. This moment is skirted by two phases, crystallization prior to choice, and clarification afterwards. Naturally, the actual selection of a course of action is important, but it takes on its fullest meaning only in terms of these two adjacent activities. In the crystallization phase of decision-making, the individual assesses the alternative paths of action. What are my chances? How have others like me managed? How much of a risk is really involved in each case? The products of exploration are worked over, pursued, refined, and sifted, until, finally, a selection is made. Then comes clarification, a more extensive following-through of the chosen path to see more fully what one has committed himself to, and to test how strongly he maintains his choice. Naturally, it is not a one-way road from exploration, to crystallization, to choice, to clarification, nor are they in fact as clearly separable as the paradigm might suggest.

Throughout the individual's passage from point to point in the decision-making process, he continues to engage in the act of turning data into information. This is a major concern of the project, since, in the real world, data are never complete. Often, it is precisely this incompleteness that makes decisions necessary in the first place. In any event, the quality of the choice depends upon the quality of the data. Before one attempts to make a decision, therefore, he must first understand the incompleteness of the data with which he is dealing.

Accepting data on these terms leads naturally to the condition that one is more likely to take responsibility for the choices he makes, since they are not totally determined by external factors. If they were, then choice would be either irrelevant or superfluous. Furthermore, in order to create information on which to base decision, one must actively process data rather than passively be guided by them, and therefore, the individual must become a significant agent in the choice process. That is, the incompleteness of data implies that the individual is responsible for his decisions in both meanings of the word: he is the one who makes the decisions, not someone or something external to him; he is the one who enjoys or suffers the consequences. This is one way to define "freedom" and it is to this notion that the project is dedicated. It will achieve this goal by developing in the student the ability to engage in this kind of decision-making relative to his career choice. That is, the project will place the student among resources, enhance his access to them, teach him the stages in decision-making, and have him engage the resources in a controlled setting so that he can develop the skills of processing data and making decisions.

An additional factor in the decision-making procedure which this project proposes is called monitoring and consists in keeping track of the student as he goes from stage to stage through the paradigm. Aside from the usual reasons for monitoring a student's behavior--to analyze his performance, select from alternate courses of action, and generally maintain an account of his interaction with a system--the project expects to present to him the facts of this monitoring so that he might use them as additional data. These facts become a kind of meta-data which the student processes. The idea of data and meta-data is analogous to the philosophical notion of being and becoming. Not only does the individual act but he becomes aware of his pattern of action. The desired result is a higher order of understanding of both the decision-making act and the panorama of career choice in which decision points are linked. Career becomes a time-extended set of choices, and decision at any given point is enhanced by an overall awareness of the road being travelled.

What the project proposes, then, is a model of decision-making behavior which requires a setting capable of providing feedback. It is an interactive setting in which an individual engages a data base in certain specifiable ways as a means of determining alternatives and selecting from among them.

Once we recognize the obvious fact that data are never complete, it becomes wise--often vital--to place the condition on choice that it be made with the best possible data available. We must ask of the data: Are they accurate? How complete are they? Do they reflect the full complexity with which we must deal? Can we get them in time to explore alternatives adequately? A library is unsatisfactory in this area, because the time involved in searching is often more than the individual can afford. Certainly large amounts of data--occupational descriptions, for example--can be stored, indexed, cross-referenced, and made generally available in a library, but that is only part of what is needed. The computer, on the other hand, is capable of all this and of providing fast access so that search time need not hamper decision-making. Furthermore, the computer can interact with the student and thereby help him to ask relevant questions about the world of work. The project looks to the computer, therefore, as a device to store large amounts of occupational data and to make them immediately and selectively available to the individual as he proceeds through the decision-making process. With this kind of accessibility, the individual can feel he is among resources and as he becomes more integrated into the reckoning environment, the data become more like extensions of him and less like external quantities.

The area of decision-making is the second one which the project sees to be dependent upon the use of a computer. As the student goes through the various stages of decision-making, he must be provided a number of things, three of which are particularly important.

First, he must have the opportunity to browse through a large set of sources and to obtain cues from the system as he goes. Browsing is not an aimless activity, it is an unstructured one, and the system should facilitate the emergence of order by probing, suggesting, reminding, and generally being at the disposal of the individual. Second, the student must be able to follow up leads, discard them, go as far as he wishes and then start anew. There should be available to him statistical projections and estimates about himself and about the world of work with which he may compare his own interests and abilities to those of others who have taken certain courses of action, so that he can begin to assess his chances and his willingness to pay the price involved in particular alternatives. Finally, he needs a way to relive his past in terms of his present orientation and to act out the future or at least to bring his awareness of the future to bear upon his current need to make a career decision. The system, therefore, must contain some kind of simulator or game player which can process student information within the larger context of information about the world of work so that the student can engage in what amounts to a "career game."

All of this requires that there be an interface between the student and the data. In many respects a human being--counselor, librarian, teacher, friend, confidant--would be the best interface, although he would not be very good at treating variables in a multivariate way or at the kinds of real-time search, estimation, computation, recording and presentation that the system requires. It is in these areas that we look to a computer, since they involve operations which a computer can be expected to do well and tirelessly. This is not to say that one will replace the other; we have no such intention, especially since neither the counselor nor the computer is totally satisfactory. They will be used together to do what they do best and thereby to function in unison as the ideal reckoning component of the desired environment. The computer then is indispensable to our effort to create an environment in which the process of career decision-making is to be maximally facilitated.

A third area of our interest relates to a new interpretation of the role of monitoring. The idea of monitoring, of course, is not new, for many people have kept track of students' behavior for later analysis. We do not limit the role of monitoring, however, to diagnosis after the fact. We conceive of a real-time monitoring and feedback system where the student processes the data from monitoring on his way to a decision. The way a person engages a subject when he teaches it differs from the way he engages it when he studies. Not only does he think more about the structure and form of the subject as a means of explaining it, but, as a byproduct, he learns something more about the subject and from a different perspective. This kind of byproduct can attend the decision-making act as well, and we intend, therefore, to provide the student with information about the structure and form of his interaction with occupational data, and about his passage through the decision procedure, so that he can use it as grist for further milling.

This, then, is the essential theory underlying the ISVD Project, although professionals in this area are sufficiently familiar with the works of Tiedeman and O'Hara to know that what I have described represents only the basic elements of the theory. I would like now to describe to you briefly what we are doing in the project to implement this theory, and while I will concentrate mv description on the computer aspects of the problem with which I am most familiar, I will first provide you with a general picture of how we have organized the tasks.

The work of ISVD is organized around three broad task areas, each of which contains two related sub-areas. The first broad area is concerned with the development of a computer-based data system. The first sub-area within this broad category consists of the collection of data on education, training, and job characteristics, and opportunities, and on the persons who will use the system. The sections of the ISVD Project that have been working in this area are the Forecasting Division, the Placement Division, the Information Division, the Inquirer-Characteristics Division, and the Data Files Division. The second related sub-area within this broad category is concerned with the development of computer programs and with the utilization of display devices to connect the user directly with the data, and this is the sole task of the Computer Division of ISVD of which I will speak in more detail later.

The second broad area of the project concerns itself with the development of a training program in vocational decision-making. One aspect of this calls for the specification and provision of the elements and process of decision-making for individuals of various ages and vocational situations. This task is being pursued by the Decision-Making Division, the Vocational Development Curriculum Division, the General Curriculum Division, and the Psychological Curriculum Division of ISVD. A second related sub-area involves supervised practice in decision-making for inquirers and counselors using the computer-based data and the computer routines. Naturally, supervised practice in the use of a computer-based information system requires that the computer-based information system exist, and since we are in the midst of creating the system, we have not yet provided extensive supervised practice, although we have carried out a number of field tests of small components of the system.

Similarly, the third and final broad area of the project which deals with the study and assessment of the system, its users and its use, will not become a major activity of the project until the system is implemented in its first prototype form.

I am sure you can tell from this very general overview of the task organization of the project that while the application of the computer to specific areas of guidance is the central focus, there is a wide spectrum of concerns and activities ranging from the development of a course in decision-making to the extensive collection and reorganization of data which the student can use in making vocational decisions. Computer people traditionally talk about software, by which they mean the computer programs they have to write to make the machines work; and hardware, by which they mean the particular configuration of machinery they are using to do a job. The Computer Area of ISVD is concerned with the problem of determining the appropriate software and the appropriate hardware for use by the project.

In describing our software concerns I find it necessary to make a distinction between what I call ISVD software and necessary-evil software. By necessary-evil software I mean those computer programs we have to write or change or adapt to make it possible for individuals to communicate with the computer while sitting at a console some distance away, and which make it possible for many individuals to be communicating with the computer at what appears to be the same time. I include in this category also computer programs needed to load data onto storage devices, to permit statistical summarization of the data and, in general, to perform necessary but highly technical and uninteresting tasks. ISVD software on the other hand consists of those computer routines that are part and parcel of the theory we are trying to implement and are integral parts of the computer-based environment we are trying to create.

First among these ISVD software components are our data bases. These are large collections of data relating to the various types of facts we intend to give students access to. There are roughly five such data bases. The first is an occupational data base which at the present moment consists of facts on approximately one thousand occupations. We have in our files as well a computer tape which contains similar descriptions for fourteen thousand occupations which we intend to add to this data base. The second data file is one we call the military data base which consists of facts for both enlisted men and officers of the major occupations in the Army, the Navy, and the Air Force. The third data file is the educational data base. Here we have data on colleges, junior colleges, and post-secondary trade schools, including where they are, who goes there, what is required for admission, the usual standards of performance expected, the careers for which one can prepare himself at each institution, and the like. We are starting out with a relatively small number of such institutions, somewhere between three and five hundred. The fourth data file will contain data on students who are using our system, other students who are attending junior or senior high school in Newton, and follow-up data on persons who were graduated from the Newton school system within the past five years or so. Finally, we will have a data base concerning personal and family living.

All the headaches that we have in the Computer Area stem from the fact that ISVD can be described as an environment in which we have on one hand a collection of data, and on the other hand, students. There are in particular three problems that we have: First, we have the problem of being able to get a piece of data quickly and easily when it is called for. Second, we have the problem of developing computer programs which will help the student associate these data as they are retrieved to make them relevant to his particular problem. And third, we have the problem of giving the guidance personnel in the project a way of specifying how these data are to be related without having them take the time and trouble to become computer programmers.

We are solving the first problem by developing a set of computer programs to permit random access to data files. These programs are being written in such a way that data can be retrieved by name rather than by address which is the usual procedure when dealing with computers. Ordinarily a person would have to know, for example, that data on doctors are stored in a particular part of the computer, and when he wanted that information, he would have to direct the computer program to go to that place and retrieve the data. Our programs will permit an individual to say, "Get me doctor," and the computer will then do what it needs to do to figure out where information on doctors is stored.

We solve our second problem, that of associating the data that are retrieved to make them relevant to a particular decision a student is making, by creating the concept of scripts. A script is a set of routines which superficially resemble programmed instructional frames which allow the student to progress from issue to issue within a particular decision, and which creates the necessary and appropriate links either to the data files, or to other scripts, or to computer programs, all of which not only help the student retrieve data but also help him decide what data he wants to retrieve.

Our third problem arises from the fact that these scripts must be written by guidance personnel since computer people, although they have the technical skills necessary to write such scripts, are not familiar enough with the guidance problem to ensure that they would be helping the student rather than hindering him in his access to data. On the other hand, it is unreasonable to expect guidance personnel to develop the extensive technical skills needed to program these scripts. Our solution to this is to create a special language which has the following characteristics:

a. It provides the means to describe unambiguously how the student's interaction with the data bases will proceed;

b. It is sufficiently non-technical, uncomputer-like, and similar to the language guidance people are used to using that it facilitates the description of scripts by people who are not computer specialists;

c. It provides all of the capabilities of existing computer-assisted instruction languages without any of their weaknesses;

d. It has extensive retrieval capabilities by being abot to call for data from our direct access data retrieval routines;

e. It has some natural language capabilities, which means a student need not give his answers by making a selection from a set of multiple-choice questions, but that he can enter his answer or request in English, and the computer will understand it;

f. It has a capability to perform statistical analysis when required.

During the past year we have created the beginnings of such a language, which we call MINORCA, which at the moment has many of these characteristics, and during the remainder of the project we will be working to improve this language so that it conforms to these requirements.

In addition to data bases, scripts, the direct access retrieval of data, and the creation of a special guidance language, we are working, as well, on the development of computer programs to permit testing by the computer and to permit simulation of career choice.

As far as the hardware, that is the machinery that is being used by the ISVD Project, we have two concerns: The first relates to the particular computer that is being used and its associated storage devices. The second relates to the terminal at which the student sits which is connected to the computer over telephone lines.

The computer we are using is an RCA Spectra 70/45. It has approximately one hundred thirgy thousand characters of memory, which makes it a medium-size machine, and it contains four disc drives, which means we can store up to thirty million characters of information which is quickly available to the computer. A data base, for example, that contained a thousand bits of information on a thousand occupations would take up one million characters of disc space, which is one thirtieth of what we have available, and so we feel we have room for a fair amount of data. Furthermore, each disc pack is removable and others can be put in its place so that our total library of data is unrestricted.

As far as the console at which the student will sit is concerned, the one that the ISVD Project will use has the following characteristics: The student will enter his answers or commands to the system through a typewriter keyboard. The computer will respond to the student through any one or a combination of the following output media: a printer, so that the student can have a hard copy of the response to take home with him; a cathode ray tube (CRT), much like a television tube, which will present textual material and simple graphical material; a random access, computer controlled slide projector which will display color slides to the student when appropriate; a tape recorder, under limited computer control; and, finally, a similarly controlled motion picture projector.

In this way, for example, we will be able to answer questions such as "How much does a carpenter earn?" either by printing out his salary, or by displaying a simple graph on the CRT showing the distribution of earnings of a carpenter, or by showing on slides a table which compares income of a carpenter with other occupations, or ideally by showing a three to five minute film which depicts the kind of house a carpenter lives in, the kind of car he drives, the kinds of clothes he buys, and the kinds of things he does for entertainment. Now, I offer this as an example, but hasten to qualify it for two reasons: First, I am not a professional guidance counselor, and thus, do not know which kind of answer good guidance would dictate; and, second, I think that while many educators firmly believe that the use of different media in the presentation of educational materials probably facilitates learning, we do not know enough of what kind of people take to what media best, or even which media are best suited to which types of presentation. One of the things we hope to learn in the ISVD Project is how we can make most effective use of various media in the presentation of data. Our hope is that at the end of the project we can specify in considerable detail what an ideal console should look like and what things it should be able to do, but at this time our guidance terminal is viewed by us as an experimental device.

Well, I have given you a cursory view of our project, describing our hopes and our intentions. I have left out considerable detail, and I have left out a number of major problems with which we are faced and which must be solved before we can honestly say that in Information System for Vocational Decisions is a workable, feasible addition to the guidance services. One major problem is whether an Information System for Vocational Decision will be economically feasible. Another is whether it makes sense to expect students to communicate with a computer through a typewriter keyboard. First, because the need to type creates certain difficulties, and, second, because it requires students to deal with their problems in terms of words whereas many individuals for which the system is devised are not sufficiently comfortable with words to make this reasonable. The third problem is that of evaluation: How does one go about evaluating an information system for vocational decisions? And related to this: How does one provide for the continual updating and evolution of such a system within the context of its widespread use in school systems?

As you can see, we are in a situation in ISVD where every problem we solve is replaced by several new ones just as important as the first. Thus, while we act as though we are moving toward a well fixed goal, and while there are times when the end is in sight, we do not have the delusion that our efforts are in any sense final.

COMPUTER MANAGED INSTRUCTION

by

Alexander Schure

INTRODUCTION OF ALEXANDER SCHURE

By Herbert Ohlman:

Dr. Alexander Schure is president of the New York Institute of Technology; he holds both a Ph.D. and a Doctor of Education from New York University. The New York Institute of Technology, a four-year college with campuses in New York City and Old Westbury, Long Island, now has a student body approaching 5,000. From its beginning as the New York Technical Institute, N.Y.I.T. has pioneered in the development and testing of new teaching methods and media. In the area of computer applications, the Institute has just been awarded a contract by the Navy to develop, test, and evaluate a computer-managed, computer-mediated, multimedia, semester course in general physics for the U.S. Naval Academy.

Prior to his presidency of the New York Institute of Technology Dr. Schure was president of the Crescent Electronics Corporation. He is a director of the Council of Institutions of Higher Education and chairman of its Educational Television Steering Committee, chairman of the Task Force on Curriculum and Development of the Electronics Industries Association, and a consultant to UNESCO and the U.S. Office of Education. He is the author of a number of articles and books, including several autoinstructional programs in electronics. Dr. Schure also holds a number of patents in the field of educational technology.

Today, Dr. Schure will tell us about the rapidly developing application of computers to instruction, and perhaps clear up some popular misconceptions about the potentials and problems of this fascinating phase of educational technology.

COMPUTER MANAGED INSTRUCTION

Dr. Schure:

Almost inevitably, when a speaker begins a presentation such as this, he castigates our American system of education in some measure--even if with delicacy. I am not here this morning to dwell at length on the various faults that perturb our schools -- in many ways they perform uniquely well. Yet, we are all aware there are problems and, indeed, that the problems are difficult to resolve. The dislocations resulting from inadequate finance, strife, mismatch of resources, or unforeseen combinations of social or political circumstances may literally be beyond the capability of a particular school to cope with solely as a single institutional system. New answers are being sought. The cycles of solutions offered are all too often ephemeral, unguided, unaccountable, or nonreplicable.

As a result of the severity of pressures, more and more institutions are beginning to draw upon resources that are outside the formal institution. The complexities resulting from management of all of the resources that surround the school involve so many factors, facts, and facets that the amount of resultant data becomes overwhelming. It is no less so with the guidance of human learning through a specific curriculum. Here, too, the educators struggle with a number of different patterns and here, the first priorities exist.

A means must become available to provide information which indicates effectiveness and efficiency of educational environments in promoting individual learning. It also is extremely important to be able to examine the wide variety of available designs on a nonemotional basis which clarifies the success of an instructional program and the decision rules governing it. Sophisticated techniques are required to do so, and fortunately they are available.

The central theme of this thesis is that through the use of the computer there can be structured more effective evaluation systems -- that we can conceptualize and interpret the more complex aspects of the educational process by use of the computer. Our discussion will focus on computer-managed curriculums. Computer management refers to a synthesis of educationally related data to yield information upon which subsequent decisions will be based, regardless of the kinds of organizations or strategies of learning examined. Thus, when we survey the conventional classroom setting, we seek all pertinent information relating to learning within this classroom. If team teaching is adopted as a strategy, its consequent result and logistics become part of the system to be managed.

Whether elementary, formal, higher, adult, technical, or continuing education is our concern, all can be encompassed and analyzed through the computer-management process. As an aside, involvement in continuing education at NYIT (New York Institute of Technology) is becoming so major that one of the wags in the research center, no longer satisfied with the concept of "womb to tomb" education, talks in terms of "conception to resurrection" responsibilities. In terms of recent biomedical advances, this is not as silly as it seems!

Regardless of the strategies and the organization of the school, computer-managed or computer-regulated instruction in essence uses the computer as the tool of the learner, the teacher, and the administrator to predict, to guide, and to yield appropriate decision points. It also allows the curriculum developer a scientific basis for development.

This use of the computer is in contrast to the methodology referred to as computer-assisted instruction or CAI. In computer-assisted instruction, the computer operating "on-line" becomes the prime resource of course material presentation, simulating the role of a teacher. CAI may be one of the modalities or one of the strategies (principal or otherwise) used by the learner, the teacher, and the school in an operating system. As used in this discussion today, CAI is only one technique, which may or may not be part of a particular institution's repertoire.

There is a growing acceptance of the potentials of new technology and media. We no longer feel threatened, apprehensive, or even uncertain about the uses of technologies with which we are familiar. All told, there are a thousand and one things with respect to complex modern technology that we readily accept. Educators can be attuned to similar comfort with the computer, once properly introduced. The errors of presenting the computer to the educational community as a teacher replacement must not be made. Rather, the device should be introduced in terms of its responsiveness to develop more adequate information relating to different judgment values and in terms of its potential to permit wide replication of desirable learner situations. Used in its capacity to permit the decision makers (the teachers, the superintendents, the local board, and the students) to be able to choose and to use wisely the best from among available resources, enormous strides forward can be made.

At the New York Institute of Technology, we have been laboring with pieces of the problem of instructional management for some years. One of the things we found (and I think my colleagues here today also find it to be so) -- is that a little bit of innovation introduced without any kind of planned base is a noisy and irregular thing. Such use tends to disrupt the school system, and very often the innovation does not become permanent. What happens is that the new process begins and exists in comparative isolation -- and then very quietly is phased out. Perhaps students or teachers do not care for it, or it might not be administratively feasible. Rarely, however, is the judgment for elimination or retention of the innovation made on the basis of truly hard data.

An early conclusion, therefore, was that any approach to long term innovation at the New York Institute of Technology must stem from an organized approach and must be part of that view which regards the educational system as a totality. Beginning in 1958, we approached the foundations. Carnegie Foundation gave us initial support funds for the design of what is now becoming -- gradually and carefully -- a complete system so far as our own institution is concerned. Subsequent major funding was also given by the Ford Foundation and recently by the U.S. Office of Education.

When I describe the New York Institute of Technology's systems, I suggest that you judge their merits in terms of their applicability to the educational situations with which you are concerned. A system, to be of value, must be highly responsive to the particular institution it serves. This is critical. Every situation has its own coping task. Princeton operates one way; Harvard has its own style; the Milwaukee School of Engineering another -- name an institution and you will find that the constituency it serves, its governing board, administrators, and faculty set up a certain kind of "task" pattern. What then happens is that the students must use their talents and the resources of the school to cope with this pattern. The New York Institute of Technology is assaying a design which, hopefully, would assist the management of any curriculum at any institution.

Recapitulating, we are then talking about a system which permits one to "manage" instruction; a system which is capable of individualizing and optimizing learning through computer-management of the educational process. To this end, let me comment on the nature of the program being developed at the New York Institute of Technology, and what kind of outputs are desired of it.

A consideration of a total computer-managed pedagogical system involves systems analysis and systems operations. The vastness of modern education dictates that the value of any new educational innovation is largely determined by its ability to integrate with other resources and components of the educational system. It is imperative, therefore, that there be a systems integration of all the available resource components, with the unique potentialities of every element aligned to produce a more effective system as a whole than is the case with resources used singly.

I. Objectives of the Project

A. To organize, develop and refine, by a systems engineering approach, a model computer-based system for the management of the educational process, which is operationally and economically feasible, and capable of:

1. Replication at alternate educational levels;

2. Self-improvement, through continuous rapid feedback of hard data recorded, interpreted, analyzed, and presented for decision by computer;

3. Individualizing and improving the learning of each student in a heterogeneous population in specified curriculums and subject sequences;

4. Managing (toward optimization) instructional programs having applicability to general and occupational education (the curriculums involved will have relevance to the grade level continuum from year 9 through year 14);

5. Establishing the parameters for guidance, counseling, and screening of students in such programs; and

6. Providing definition of goals, course content, prescriptions, and instructional strategies related to the selected curriculum areas (acceptable to senior advisory boards consisting of representative educational, management, and industrial experts).

B. To implement, test, revise, restructure, and optimize the system in selected courses (mathematics, electronic technology, physics, computer sciences) in a selected environment at the New York Institute of Technology.

C. To demonstrate replicability and viability of the system for one course (the articulated physics program) at the branch faculty of New York Institute of Technology, and at least one other environment typical of the Nassau County Secondary School System.

D. Concurrently, to train those members of the community operationally involved with the educational process (the learners, the teachers, the guidance counselors, the department chairmen, the administrators, the school board members, the trustees, and such other members of the social, industrial, and political communities as can provide additional opinions or expertise in the development and implementation of a superior learning system) at both the secondary and collegiate levels, so as to ensure effective operation of the system.

E. To disseminate to interested segments of the educational and social communities the findings and conclusions of this study.

II. Procedures

A. In order to accomplish the objectives of the project an "action research" program has been prepared, which will, when successfully implemented, achieve the specific goals previously set forth. Procedurally, the action research program may be considered to encompass several interrelated categories of activities.

B. An initial and continuing activity category covers the design and development of the computer-based management system itself, through which the assessment, revision, restructure, optimization, and validation of the instructional system is to be accomplished. Since the system is self-generating and self-correcting, these activities involve empirical implementation and observation under actual school conditions, and testing, modification, and retesting in such a viable situation.

C. Therefore, a concurrent category of activities addresses itself to the development of subject matter courses, with all appropriate support material, and their testing, revision, restructure, and ultimate optimization and validation. These activities are governed within the instructional management area of the system. The method of attack -- the step-by-step procedure whereby the courses are behaviorally defined, structured, refined, and validated -- evolves directly from the system design and is controlled and directed by the computer-based management system.

D. A third category covers the tasks involved in the replication of the management system and the teaching of the sample courses in the selected other environment, and the training of those personnel concerned -- at the Institute and among the cooperating school systems.

E. Finally, there is the dissemination of the findings of the study for broad application throughout the educational community.

F. These major categories of activities go on, to a certain extent, concurrently, and one cannot proceed to culmination without the others.

G. The course development activity, particularly as it reaches the implementation and assessment stage, depends entirely upon the management system with its immediate feedback and analysis. The management system, conversely, requires the course development activity as its "test bed" upon which it bases its own growth and shape. Final validation of both the course and the management system depends upon the experiences of the replication experiment.

H. A step-by-step outline of the procedure is set forth below, the steps including:

   1. Development of the system design for the computer-managed system. This system is presently being designed at New York Institute of Technology by the in-house staff of the Institute's Advanced Systems Laboratory. The management system encompasses subsystems. Among the present major educational subsystems comprising the model are:

      a. Goals (curriculum objectives expressed in behavioral terms delineating precisely the substance of the educational program and the skills and knowledge to be learned)

b. Students (as inputs to the system, with profile structure, academic levels, proficiency attainment, and all other relevant data relating to their selection and subsequent education)

c. Curriculum, course, and instructional content (software and programs designed to accomplish goal specifications)

d. Instructional strategies (the combination of methods, media, and organization required to conduct the learning program)

e. Assessment, tests, and procedures (ranging from student performance through program cost effectiveness)

f. Instructional decision-making and prescriptions

g. Feedback and restructure mechanisms

h. Organization and facilities (personnel, facilities, faculty, and equipment required to support other subsystems)

2. Implementation of the system at New York Institute of Technology.

   a. Accumulation of all pertinent input data appropriate to the subject matter areas and the teaching-learning environment involved, determining as much as is practicable about the students, the existing programs, instructional strategies, media employed in the subject matter areas, past performance records, etc. During this process of data accumulation, the faculty is directly involved. Specific steps include:

      (1) Primary faculty orientation

      (2) Examination of student qualifications

         (a) Prerequisite knowledge base, scores as available in basic knowledge areas, recency of learning experiences of prerequisite knowledge

      (3) Examination of prior performances of student in the major subject area categories

         (a) Analysis . achievement test scores in each defined area

      (4) Discussion with faculty to:

         (a) Define level of comprehension ultimately to be achieved by the student upon successful completion of the course and in each major subject matter area category

(b) Define existing overall course objectives

(c) Identification of current behavioral objectives, if any

(d) Identification of existing remedial prescriptions

(e) Review of existing instructional strategies, order of presentation (sequencing), and media and teaching aids employed

(f) Review of existing tests and assessment

(g) Review of existing achievement criteria

b. Analysis of all input data thus accumulated, and (giving due respect to faculty desires) the defining of suggested changes and additions to existing course objectives, particularly by determining areas in which improved quality of student performance (or reduction of time while maintaining acceptable quality) is clearly indicated

c. Development of modifications and additions to existing course content, including new overall course objectives

d. Preparation of behavioral objectives. When all the above detailed steps necessary for the development of course content, course objectives, planned remedial areas, etc. (the syllabus) have been accomplished, the researchers then redefine the detailed behavioral objectives necessary to fulfill the syllabus, taking into account:

(1) Level of instruction desired

(2) Total time allowed for the overall course

(3) Specification of enrichment or optional behavioral objectives

(4) The requirement that the sum of all the behavioral objectives in each subject matter area cover the topic to the level desired by the faculty

e. Sequencing the courses considering such factors as:

(1) Existing student knowledge and skills

(2) Interaction of subject matter content

(3) Sequencing based on historical patterns (population stereotypes)

(4) Instructor and/or learner preferences and biases

(5) Difficulty of material

(6) Availability of media

f. Development of the proficiency measures to:

(1) Expose areas for remedial branching

(2) Establish the extent to which behavioral objectives will have been achieved

(3) Permit assessment of learner or class achievement and establishment of learner or class standing

g. Selection of media of instruction, identifying resources, and determining instructional strategies, based upon the following criteria (some of which are particularly pertinent to this program, in view of the flexibility and ease of modification available in a computer-managed instructional system):

(1) Success (or failure) of existing resources and strategies as currently used at New York Institute of Technology and at similar institutions. (Effective use is to be made of those resources that have been proved to be successful in related programs.)

(2) Ease of preparation and relative cost

(3) Ease of modification

(4) Applicability to evaluation or scoring

(5) Uniformity with which techniques can be applied. (Can quality control be maintained? One of the advantages, for a computer-managed instructional system, or of pretaped ETV, over a live classroom presentation or demonstration, is that in analyzing and evaluating the results of the instructional program, the noncontrollable variable of teacher performance is eliminated.)

(6) Ease of transition to remedial branches

(7) Consistency of media, as applicable to a sequence of learning units.

(8) Ease of transition from one media to another

(9) Naturalness (for example, using audiotapes, if the learning problem involves differentiation of sounds)

(10) Need to sustain interest

The correlation of media selection and instructional strategy planning is very close, and all the above criteria may well be applied to this decision-making area. In the course of previous instructional development programs with which the Institute has been involved, including on-going internal efforts, the overriding criteria for the development of instructional strategies is the simple accumulated judgment of the best way to get to the goal; i.e., to achieve the behavioral objectives.

h. Provision of initial instructional materials, choosing from existing appropriate resources or producing new materials

The computer-management aspects of this program presuppose test and assessment of all elements of the instructional process, and revision, restructure, retest, and further restructure on a continuing basis. This methodology dictates that the instructional materials, particularly in their early generation, be as inexpensive and suitable for modification as is consistent with the minimum effectiveness required for evaluation. If, for example, a motion picture is judged to be the optimum device in a given situation and no appropriate film is available from existing resources, it is initially produced by the "minimum film" process; after evaluation and analysis of the content presentation has indicated needed revisions, and these, in turn, have been made and have been tested, the film can be remade with appropriate technical perfection.

3. Assessment, analysis, and revision of the instructional management system.

Critical to the success of the computer management is the Automated Adaptive Feedback Subsystem (AAF) developed as part of the management system (see Figs. 1-3). In essence, AAF is a self-adaptive system embodying prediction, evaluation, record-keeping, and feedback mechanisms. The objective of the AAF subsystem is to provide a logical basis for simultaneously improving curriculum and optimizing instructional decision processes for increasing individual student performance.

Perhaps the single most important design characteristic of the proposed subsystem is its property of continual improvement. This allows for continuous testing of new ideas in course presentation and content. In addition, the system continually improves its prediction capabilities relating to individual student instructional strategies. The proposed AAF subsystem can be the kernel of a much more extensive feedback system relating academic and post-academic (occupational) functions.

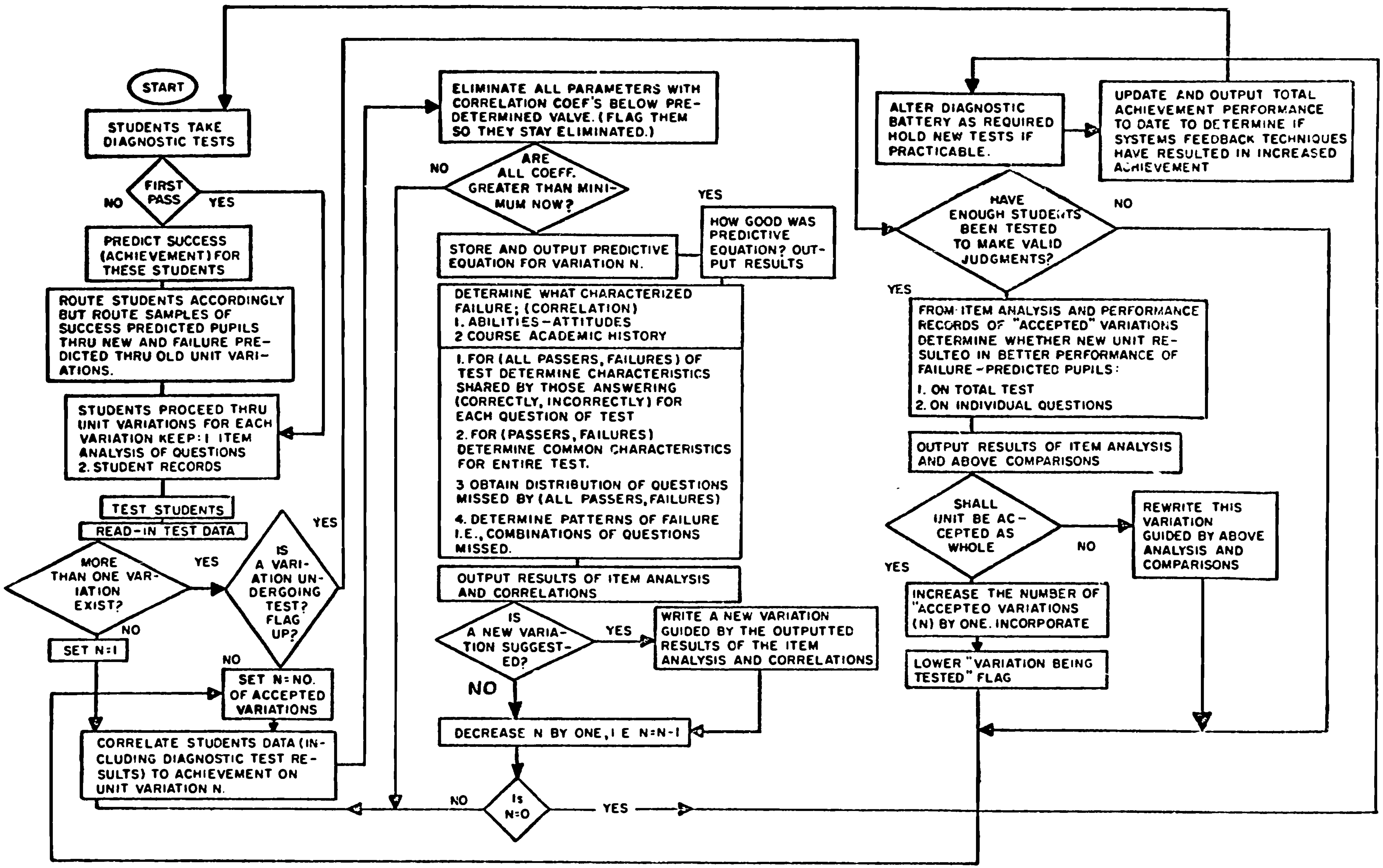


Fig. 1: Preliminary Flowchart of Automated Adaptive Feedback Subsystem

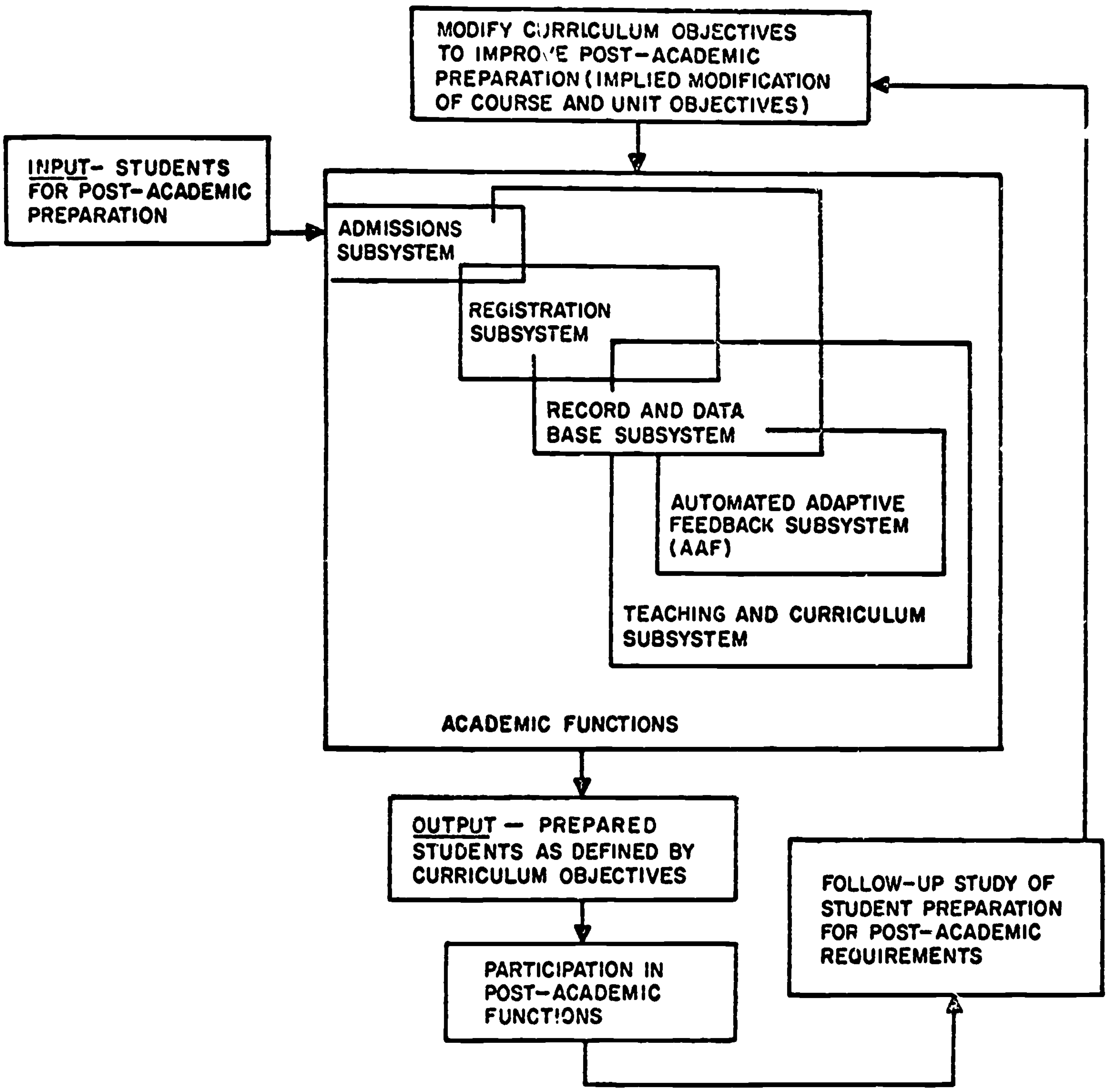


Fig. 2: Flow Design: AAF Subsystem as Kernel of Extensive Feedback System

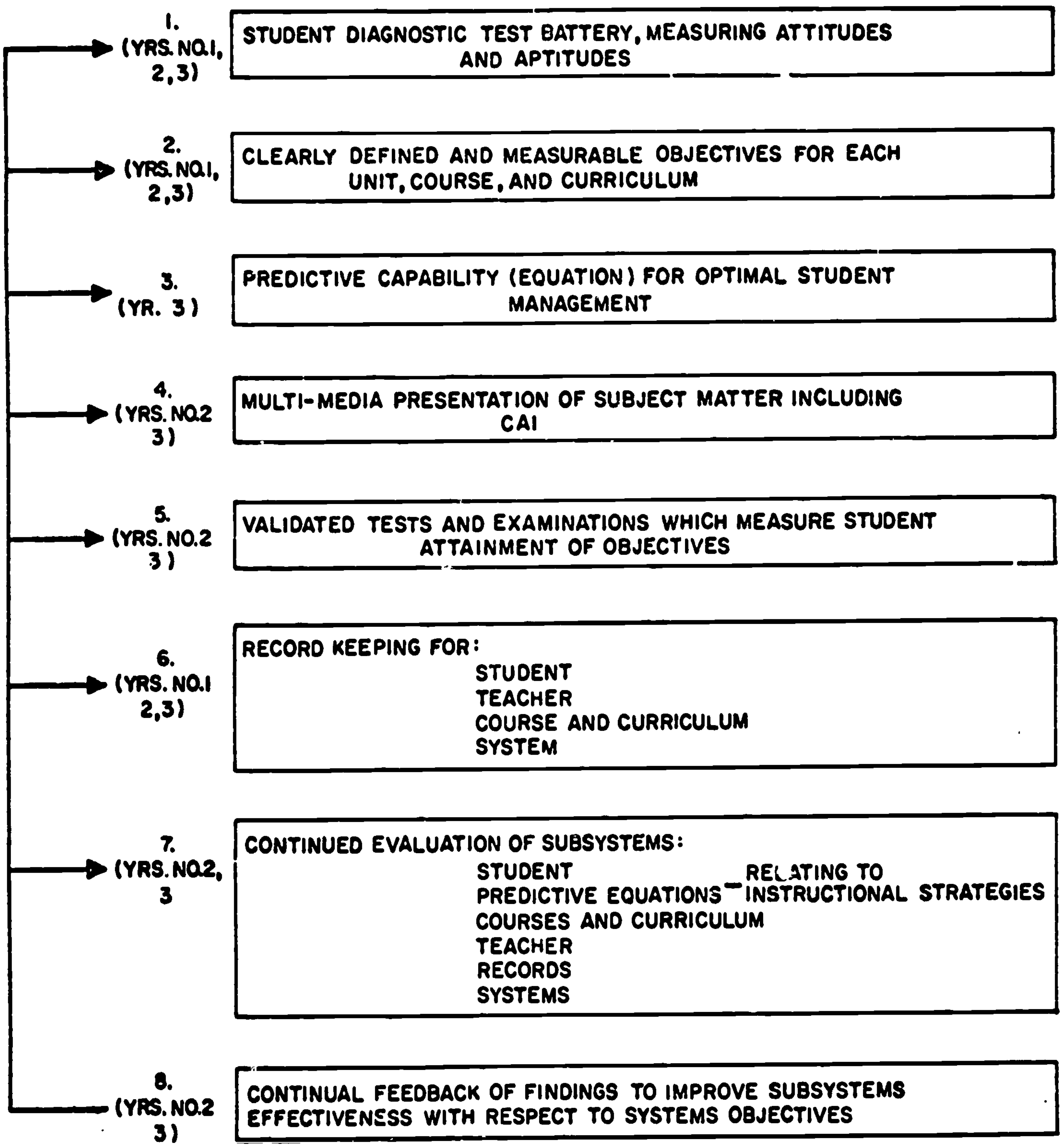


Fig. 3: Salient Elements of the Proposed Adaptive System

Although extensive analysis and design will be required for this subsystem, the following requirements seem evident:

a. Design feedback mechanism

b. Establish appropriate diagnostic test battery

c. Quantify measurable objectives for units, courses, and curricula

d. Establish appropriate statistical techniques as the basis of the decision-making processes relating to:

    (1) Choosing apprc, riate samples

    (2) Correlations between student data and

        (a) Achievement

        (b) Learning efficiency

        (c) Retention

        (d) Prediction

    (3) Criteria and testing for superseding existing equations, units and diagnostic tests

    (4) Item analysis for unit questions

    (5) Criteria and testing for evaluating effectiveness of feedback system design

e. Validate tests and examinations as truly measuring objectives of system

f. Program feedback mechanism

g. Incorporate record-keeping capability outputs

h. Program statistical algorithms

i. Integrate total systems programs

Alternative means by which the systems functions can be accomplished will be investigated and developed preliminary to the final systems evaluation.

The implementation of the managed instructional program in the four subject areas, their restructure, and the optimization of design, methodologies, and resources initiated during the first year will continue as an on-going phase of the program.

The research design of the program for testing the educational efficiency of the instructional management system is set forth in a separate Institute publication. This model has been successfully employed in previous studies at New York Institute of Technology, and, although originally developed for the general physics curriculum, is equally applicable to the electronics, mathematics and computer courses.

A procedure initiated during the first year will be the use of a committee of appropriate subject area teachers and consultants from other environments, working with the subject matter specialists from New York Institute of Technology in the articulation of the new programs, particularly in regard to the upper high school and lower college levels of the mathematics and physics courses.

A feature of this organizational pattern will be the opportunity for the teachers and the consultants to test out a daily application of the evolving system in the classroom. Necessary modifications can take place immediately in this way, thus eliminating the necessity of waiting still another year or longer to make essential changes. Thus, short term as well as long term feedback mechanisms are on-going elements of the system.

Data will subsequently be processed to evaluate the validity of test questions, to permit the teacher to have an in-depth evaluation of the effectiveness of learning or teaching per unit, and to indicate certain areas of instruction for retention or rejection and emphasis or de-emphasis in future courses (see Fig. 4). These data will also permit the keeping of a history of each student so that his progress in the course can be charted and observed for guidance purposes, his fundamental areas of strength or weakness can be recognized, and appropriate supplemental instruction can be provided.

Procedures for restructure and optimization initiated during the first year will continue throughout the entire project, increasing in scope and value as the data base is augmented and the systems design improved.

4. Orientation and involvement of those people concerned with the educational process in the environments under consideration, and training of the operational personnel.

This activity is on-going throughout the program. Continuous contact on an almost day-to-day basis will accomplish much of this objective. Certain structured procedures will, nevertheless, be instituted.

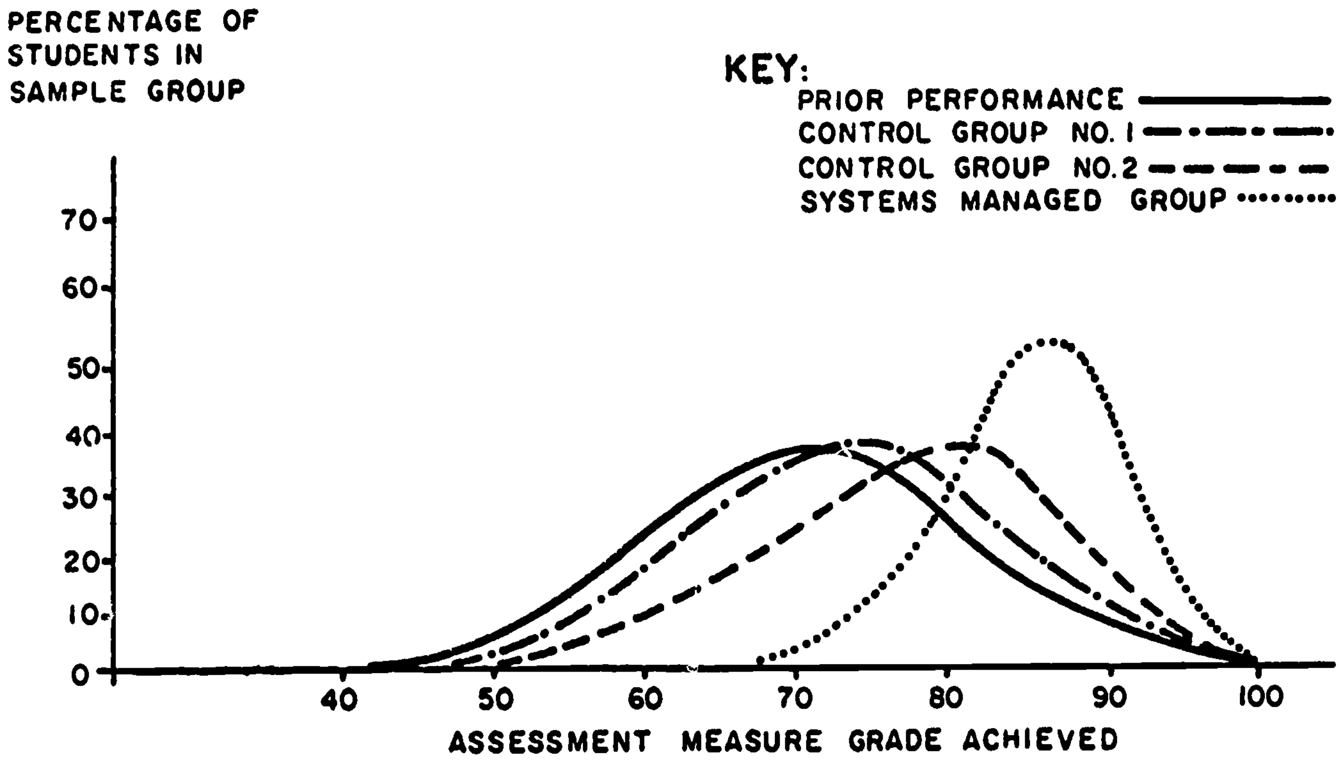


Fig. 4: Overall Course Objective Achievement, Grade Distribution

---

Type of chart to be prepared for graphic demonstration of students performance, etc. (Examples are purely hypothetical and do not represent actual data).

The orientation programs and work programs are considered a vital part of the project. The New York Institute of Technology has presented programs of this type in the past with the Mineola School System, the Wantagh School System in Nassau County, the Brevard County School System in Florida, the Anne Arundel School System in Maryland, and for the Committee for Cooperative Action, representing a cluster of seven districts in Nassau County. Furthermore, personnel at the New York Institute of Technology have acted on a subcontractual basis as consultants for The Educational Council, representing the 57 districts of Nassau County, and have presented materials before committees outlining methods and resources as well as explaining instructional management as orientation for the members.

The activities planned for the proposed program (which are similar to the previous successful experience) include presentation sessions at least once a month for participating and cooperating school staffs. In addition, workshop sessions are provided in which participation by the individuals is required. These sessions will cover topics such as systems and management concepts, data processing, etc. They will also include presenting relevant new concepts to educators (for example, demonstrating the utilization of the computer as an important tool in decision-making for administrators, teachers, guidance counsellors, and students).

5. <u>Dissemination of the findings of the study</u> for broader application of the system and subsystems.

The presentation of the findings is to be disseminated in the professional literature and appropriate professional congregations. The Institute staff will also offer its services for consultation to interested school systems who might wish to apply the system in their own settings, and will operate ongoing seminars for appropriate professional groups.

The Institute will publish a detailed final report, to be made available as appropriate through the U.S. Office of Education.

III. Desired Outcomes

The demonstration of the effectiveness of just-outlined "System for Individualizing and Optimizing Learning Through Computer Management of the Educational Process" will have pertinence beyond the limits of mathematics, physics, electronics, and computer programming at the college level. The system will have applicability to a wide range of grade levels and a broad spectrum of subject areas. It should be understood that it is the primary purpose of this project to study the viability of such a management system; the development of the specific course content and format is secondary, although the products thereof will be substantial, and capable of replication in other learning situations.

The experience gained in introducing change in school systems should be helpful to other innovators. The program is aimed at arriving at a better learning system than is presently in operation either at New York Institute of Technology or in the cooperating school systems. The program should result in better organized, accurate, hard data relating to the materials, equipment, and systems that will be used in actual instructional operations. It will certainly provide for testing and assessing the various instructional methodologies employed, as well as verifying the content and the performance of the software and strategies used to obtain any particular learning point. It will also provide a frame of reference, and assessment data relating to a set of strategies, a set of instructional materials, and a set of hardware suited to the four subject areas under consideration, and relevant to the needs of particular students undergoing these learning experiences.

The professional teams concerned with this program will have integrated into the instructional process an extensive range of media, selected from among all the texts, films, television tapes, audio tapes, transparencies, slides and filmstrips, workbooks, and behavioral games available today, and any other materials that are necessary to attain the educational aims of the courses, and to design new aids and devices when needed.

Computer-assisted instruction as a technique will have received due attention. It is intended to experiment with CAI as a prime resource in the computer sciences program. The objective here is to provide experience in the use of CAI (and to evaluate its effectiveness) for the teachers and students in the Institute and in the cooperating secondary school system.

The computer-based analysis and evaluation of the results of the use of this range of relevant material should enable a teacher to move from the old pattern of selecting from a variety of uncoordinated materials to a coordinated integrated system capable of easy updating, and based upon a store of specific experience about particular learning materials, equipment, and software demonstrated to be pertinent to the subject matter areas treated experimentally in this program.

To summarize, among the desired outcomes of this proposed research program will be the development of a model computer-based educational management system that will have improved individual student learning; that will have been demonstrated operationally feasible and capable of replication in other educational environments; and that will have yielded data serving to demonstrate the validity and cost-effectiveness of such a system.

Courses in four subject areas, optimized as a result of data feedback and analysis, will have been developed, and training of teachers and other participating specialists, at New York Institute of Technology and the secondary school educational environment in which the system will have been replicated, will have been accomplished.

DISCUSSION: ALEXANDER SCHURE

Question: I would like to know if you can give us an example -- you can use any one of the curriculum areas -- about how the system is capable of adjusting itself.

Reply: Let me answer this question by describing the philosophy and guide lines which we used to govern the development of the New York Institute of Technology Transitional Physics Course.*

Question: You mentioned that you are really working on systems for urban and suburban uses. What are we doing for small, isolated rural schools with these kinds of programs?

Reply: Eventually, centralized computer time-sharing facilities from organizations such as CEMREL or NEEDS will answer this question. The time is coming when access to data facilities will be available to a rural area without the need of the local constituency to maintain a supporting staff. The probable links to the central computer could very well be touch-telephone input, some kind of data transmission device, and an output device as simple as a high-speed teletypewriter. Through conventional telephone lines, the rural schools would be able to access a support facility now not available to them.

Question: I am interested in the composition of the group of people that is necessary to produce an operating system such as you're talking about.

Reply: The New York Institute of Technology has about ten people in various categories of support, including senior analysts, senior programmers, junior programmers to provide input of information to the computer, and two junior system analysts (as a minimum) working with the educational teams.

Question: What kind of people are they? Where do they come from?

Reply: All kinds and types. The educational community draws them from wherever they can be found. Some are university trained, and others come from the business or industrial fields with prior experience in machine techniques. A larger cadre of qualified people are becoming available from the increasing number of university and private school programs related to various aspects of computer science. The New York Institute of Technology has its own computer science program which helps augment its personnel bank.

Question: You mentioned programmer availability. What about the availability of the systems analysts? What are we doing to develop these people?

---

* A full description of this course is given as an Appendix to this discussion.

Reply: Substantial numbers of top systems analysts have not been readily attracted in significant measure to the educational institutions, mostly for reasons of institutional finance capacities. Matching industry's salaries is difficult for the average institution.

Question: What would be necessary to take this system out of your institution, lift it, move it to another institution, and plug it into the proper hardware to make it work?

Reply: Time and training of staff has taken a year and a half to replicate one system, primarily because what's plugged into the system has to grow out of the user school itself. The original system is a guide and a skeleton. In the replication, it is adapted to the prime needs of the school. The most important element in transferring the system is provision for adequate training for those who will be concerned with its operation. In Nassau County, replication is based on a lead time of a year to make the system fully operational.

Question: Are the programs available?

Reply: I think they will be. Some of the work that we're doing is in the public domain. Many programs are legally available, but often there tends to be a proprietary withholding on the part of the staffs that have developed them. But I would say that your answer, within the next years, would be yes.

Question: Don't you think that (in the answer to the previous question) many of the people in those programs in the schools (after they've received this instruction) will become readily available?

Reply: Yes, I'm sure that they will. My reservation occurs because I thought that the question related to senior systems analysts who are capable, for example, of building a mathematical model of complex systems. They're not so readily available. I think that more of them will be coming along soon. There should not be a staffing problem within a few years in the lower level of systems operation personnel. There is little difficulty in getting people to program or run your 1500 system. For routine work, many professors and teachers have the capability and talent. This is less true with the very sophisticated analysis, I've found.

Question: Maybe you'd like to indicate the kinds of educational backgrounds these very sophisticated men might come from?

Reply: They vary, but most have a highly mathematical orientation -- usually from majors in physics, mathematics, or related fields. We're talking about the upper systems analyst levels. Let me insert a caution here -- not all of the analytical requirements to solve an educational problem are so complex as to require this kind of person. You will be able to find, graduating from many institutions, an increasing number of students perfectly capable of designing excellent operational systems. Still, for the discernable future, people of top capacity are going to be in very high demand. The market is open to them, their opportunities for choice are diverse, and very often they do not enter the educational world. It is just as exciting to work on operational analysis for IBM, General Electric, or Honeywell, with their attendant benefits and status, as it is to work for an educational institution -- and, unfortunately, often more lucrative.

APPENDIX: NYIT TRANSITIONAL PHYSICS COURSE

Scope, Objectives, and Media

The NYIT Transitional Physics Course is based upon many of the recommendations of the Physical Science Study Committee as set forth in the text published by Heath and Company, as well as material outlined in Physics, Fundamentals and Frontiers by Stollberg and Hill (Houghton-Mifflin), and essential elements culled from the New York State Regents Syllabus for Secondary Schools, as well as the considered consensus of the NYIT Physics Department faculty (see "Analysis of Subject Matter" below). Most of the subjects have been increased in depth sufficiently to validate the transitional nature of the course.

The general instructional objective of the course is that the students who complete it meet the NYIT Physics Department's minimum performance requirements for enrollment in the Institute's three-semester college course.

The scope of the course includes a range of information and skills, beginning with significant figures, scientific notation, and measurement, through all of mechanics (statics, kinematics, dynamics, heat and temperature, geometric and physical optics, electricity and magnetism, wave motion and sound). It does not include modern or nuclear physics (these topics are to be developed at a later date).

Instructional media include the Autotutor and its intrinsic programs, audio tapes, slides, single-concept 8-mm films, sound movies, video tapes, self-operated demonstrations, and instructor-supervised laboratory exercises. In addition, one or more instructors are constantly available for individual student conferences.

The fundamental objective of the multi-media articulated transitional course is to teach a greater number of approved topics in a shorter length of time than has ever been possible by conventional methods, and to permit self-pacing according to accepted standards.

METHODS OF DEVELOPMENT

Analysis of Subject Matter

Analysis of subject matter by empirical methods involving various evaluative techniques has resulted in a course curriculum couched in terms of behavioral objectives. The behavioral objectives were initially developed by analysis of a broad base of relevant authority and empirical material.

First, a review was made of all pertinent sources, including the findings of the New York State Committee for Review of Secondary School Physics curriculum, the recommendations of the Physical Sciences Study Committee (PSSC), and the examination of presently existing approved tests and assessment measures in parallel courses, such as the New York State Regents examinations. Then a consensus of the Physics Department Staff of the Institute was derived, on the following criteria:

1. Examination of student qualifications (prerequisite knowledge base)

2. Prior performance of Institute students (analysis of achievement test scores in each defined area)

3. Level of comprehension ultimately to be achieved by the student upon completion of the course

4. Existing behavioral objectives

5. Existing instructional strategies

Assessment of each objective in terms of student performance has been in process for some time and will continue in the future. It is anticipated that the application of several methods of logical analysis as well as empirical evaluation will result in a final set of behavioral objectives which describe a rich and rewarding course in transitional physics, yet one which is free of repetitive materials or subject matter segments that are already part of the student's stock-in-trade.

The preliminary complete Autotutor course has already been printed on 35-mm film, based upon the initial set of behavioral objectives generated by staff analysis of PSSC recommendations and modern texts as noted; in addition, some assessment work has been done on these objectives with the help of a limited number of high school students who volunteered their services during the summer. The empirical evaluation techniques will be continuously applied until the final course of study is completely evolved.

Classification of Objectives

Behavioral objectives for a discipline such as physics (also applicable in other fields) may be classified functionally to good advantage. Any given objective may have as its outcome behavior which indicates that the learner has fulfilled the following outcomes (all outcomes are not necessarily exhibited by every learner):

1. Has committed an essential tool fact to memory and can reproduce this fact at will

2. Understands and can reproduce a principle or law of general nature

3. Can apply any single principle, law, or concept to the solution of a single-step problem, descriptive or numerical, in which direct substitution is indicated

4. Possesses the requisite basic mathematical skills to convert an equation or proportion on the desired level to a form that is usable in solving a single-step problem in which direct sbustitution is not possible, as in this example:

Given that centripetal force, $\underline{F} = mv^2/\underline{r}$, find the tangential velocity of a body in a circular motion in which $\underline{F}$, m, and $\underline{r}$ are known (this requires the ability to solve the equation literally for $\underline{v}$)

5. Can make decisions involving the order and handling of sequential steps in the solution of a multistep problem (descriptive or numerical) selecting and evaluating the required principles

6. Can make decisions which permit him to discard "false clues" and select for use only those factors which have a bearing on a problem situation

Attention is called to the underlined phrases in outcome behaviors 5 and 6. It is believed that decision-making behavioral objectives in physics represent the essence of true education in the subject (or for that matter, in virtually any field). The realization of decision-making objectives demands that appropriate attitudes and methods of attack be developed as necessary for each individual learner. It is in this area where self-pacing plays its most important role.

The decision-making objectives can be subclassified in various ways. In general, for physics, the following subclassifications appear to be valid:

Given a multistep problem situation:

a. The learner must be able to list physical relationships which may possibly have a bearing on the solution

b. He must be able to discriminate between relevant and irrelevant data, and immediately discard the latter

c. He must be able to organize the relevant data and applicable relationships, one with the other, in logical order for subsequent handling

d. He must be able to fuse data and relationships sequentially to arrive at the correct result

Category Method of Modifying Initial Behavioral Objectives

The initial behavioral objectives are those drawn up by the process described above. The objectives are next to be sorted by classifications 1 through 6, including subclassifications a through d. Sorting by category provides a natural means of checking for omissions and repetitions, particularly the former.

The objectives for each topic (e.g., The Law of Universal Gravitation) are examined to determine if objectives in every classification have been included. A matrix may be set up in which subtopics appear along one axis and categories along the other. Each cell of the matrix may then be examined for omissions or repetitions. It is felt that a large number of behavioral objectives may be added in this manner.

Hands-on Behavioral Objectives

Distinct from the academic objectives thus far discussed are the equipment skills which must grow as the educational process progresses. Not only is laboratory work an essential part of the physics program in the total educative picture, but it also offers a new and different set of criteria upon which assessment of the success of the program can be partially based. Such objectives are initially drawn up in a manner similar to that used for the academic goals (from faculty and student desires, and by means of the appraisal of existing laboratory exercises and judgment as to the suitability of each experience to the overall goals of the course). Evaluation of such material is to be made by members of the Physics staff for the purpose of preliminary selection, and is later to be improved and modified by observation of individual students under laboratory conditions. A check list in the hands of a laboratory instructor should quickly disclose weaknesses and shortcomings in any exercise, and permit alterations necessary to perfect it:

Check List

1. Can the student understand and follow equipment set-up instructions?
2. Is the equipment suitably prepared to avoid fumbling and wasting of time?
3. Are the performance instructions sufficiently clear so that time-consuming errors are eliminated?
4. Are the measurements expected of the student easily obtained with the anticipated precision, using the equipment with which he has been provided?
5. Have provisions been made so that the student takes appropriate notes?
6. Can the experiment be performed within the established time limit?
7. Do the results justify the performance of the experiment?
8. Does the exercise succeed in teaching the point or points for which it was designed?
9. Does the exercise tend to add to the equipment skills already possessed by the student?
10. Can the student write a meaningful, complete report on his work?

From this check list, the specific hands-on behavioral objectives of the course are derived. For example, the first check list item translates into:

> "Student must set up the following equipment (specifications follow) in x time and under x conditions."

Empirical Pretesting of Objectives

In this phase of course design, a series of questions which relate to the objectives in a given topic in physics are intuitively generated.

There is at least one question per objective, the question being phrased as best as possible to ascertain the level of the individual student's background.

The objectives are then refined on the basis of the data obtained in two basic ways:

1. Questions are revised to eliminate difficulties which keep them from measuring the behavior they are intended to measure.

2. Objectives are dropped when they are shown by the data to represent material already known by the student population.

The students to be used in this type of test are those who have had a previous course in basic high school physics, and at least one course in integrated high school mathematics, involving some algebra, geometry, and where possible, trigonometry. Although it is recognized that heterogeneity of previous accomplishment will be present, it is held that such pretesting can identify very definitely those objectives which may either be eliminated entirely, or relegated to home-study for the purpose of releasing time for more essential classroom teaching.

The same procedure is to be repeated for all the topics in the total course. Similar procedures are utilized to establish and verify necessary prerequisite behaviors to insure student capacity to perform in the curriculum under design.

Empirical Design of Instructional Material

Thus far in the development of the transitional course and as far as can be foreseen in the future, the initial selection of instructional material and media will be based on these few simple maxims:

1. Don't use instructional materials unless previous tests have shown that the student needs them.

2. Select and sequence various media on the basis of their natural application to the subtopic at hand:

   a. Slides are ideal for visualizing static equipment, equations, diagrams, etc.

   b. Single-concept films and sound movies are suited to dynamic situations in demonstrating sequences of physical events.

   c. Audio tapes economically interject the personality of a master teacher.

d. Printed worksheets which accompany audio tapes provide the medium for interaction between student and teaching devices.

e. Video sound tapes (for closed circuit television or film) bring the master teacher right to the student's desk.

3. Each sequence of instructional media for every individual subtopic in physics is to be followed by a progress test. This test is to be based on categorized behavioral objectives as previously modified and optimized.

Evaluation of Program Effectiveness Using Progress Tests

Progress tests take the form of multiple-choice question groups, each question being coded to a final category designation of its parent objective. Answers to questions will be entered by the student on a computer input form, the computer being programmed to grade the paper, and feed back information which the student can use to perform remedial tasks. This information will consist of one or more of the following suggestions:

1. Assigned reading of specific pages and paragraphs in an approved text.

2. Repetition of specific sections of the program just completed.

3. Listening to remedial tapes or viewing remedial films.

4. Assigned conference with an instructor or a seminar.

5. Alternative instructional strategies as appropriate.

Simultaneously, the computer will be programmed to generate student profiles for access at any time, and also to build up an evaluation of the clarity and effectiveness of every question used in the progress test. The validity of a question will be determined statistically by comparing overall student achievement with achievement on the specific question. It is to be recognized that the repeated failure of a given question may be due either to the invalid nature of the question, or to the ineffectiveness of the program approach, medium, or sequence.

In revised designs for assessment, any desired format of progress test or assessment of achievement can be utilized through use of computer-assisted instruction, or by data assessment by peer or professional (subsequently transferred into format for referral, by computer or other means, on the basis of such achievement).

The assessment measures are regularly reviewed to insure that they are actually measuring the student's ability to perform at the level specified by the behavioral objectives.

For such questions (of which there will be many), initial modifications of presentation will first be made if the staff feels that the question itself is reasonable. Modification of the question will be made if successive changes in presentation do not significantly alter the apparent failure of the question to elicit the correct response.

Additional progress tests of equivalent difficulty and coverage will be prepared for those students who have been assigned remedial work. The same evaluative methods will be used for the second test in each topic by using it as the primary test for alternate groups of students, as each group reaches the topic in question. By having two or more sets of progress tests for each topic, the integrity of the examination will be better preserved.

Controlled Testing: An Empirical Evaluation of the Entire Course

Over a period of not less than three years, physics material identical with that contained in the multi-media course herein described will be taught by the conventional lecture-recitation-laboratory method to a control group which will never number less than 20 students per semester. Using both standard academic testing procedures and laboratory performance exercises, comparisons will be drawn between the students in the multi-media course and those in the conventional course, for the express purpose of locating and identifying weaknesses. Included in these measurements will be comparisons of course length, mean scores, standard deviations, lowest score on each test, highest score on each test, the range of the scores, the number of students involved in each case, and such additional correlations as are deemed advisable by faculty and behavioral psychologists concerned in the course.

Organization and Facilities

A number of preliminary steps have been taken to organize a training center for instructors who are to participate in the development and growth of all types of multi-media courses as planned by the New York Institute of Technology.

Mechanisms for training teachers in the use of the various media, and in the generation of these media are now being set up.

The basic facilities for preparing the required audio tapes, slides, single-concept 8-mm movies, sound movies on 16-mm film, and video sound tapes have already been installed on the premises. For computer assisted instruction, an IBM 1500 is due to arrive during the early part of October 1967. In addition, IBM 1620 and 360/50 configurations are available, and have been programmed to support the instructional systems. The growth of faculty and facilities to meet the demand of the increasing number of multi-media now available, or past the planning stage, will be carefully supervised by experienced members of the staff of the Institute.

OPERATION COMPU/TEL -- REMOTE USE
OF COMPUTERS BY HIGH SCHOOL STUDENTS

by

Peter G. Lykos

INTRODUCTION OF PETER LYKOS

By Herbert Ohlman:

Our next speaker is Professor Peter Lykos. Dr. Lykos is professor of chemistry and director of the Computer Center at Illinois Institute of Technology. This center comprises five departments, but I'll let Pete tell you about these because I'm a little bit confused about them myself, and how they managed to get five whole departments into a little computer center!

Professor Lykos earned his Ph.D. in theoretical chemistry from the Carnegie Institute of Technology, and has contributed a large number of papers to his primary field, which he is currently teaching at IIT. He is also chairman of the National Academy of Sciences' Committee on Computers in Chemistry, a recent chairman of the Association for Computing Machinery's Chicago Chapter, and past chairman of the Chicago Chapter of the American Chemical Society's Educational Committee.

Today, I hope that he comes to us wearing his computer hat. He just told me that he had been speaking for two hours last night on computers in chemistry, and I was hoping he wouldn't get confused and give us the wrong talk, but he says fortunately he has left his slides home so he can't do that.

Dr. Lykos is going to tell us about his pioneering efforts to introduce computer science into the high school curriculum, which comes under the name of Operation Compu/Tel. CEMREL has been able to reprint a detailed report on this secondary school program by his colleague, Mr. Anthony Peluso, who is chairman of the mathematics department at Lane Technical High School in Chicago. Mr. Peluso's paper follows the discussion; a summary of it has been published in American Education, May 1968, under the title "An Answer for a Dime."

OPERATION COMPU/TEL -- REMOTE USE OF COMPUTERS BY HIGH SCHOOL STUDENTS

Professor Lykos:

Thank you Herb. It is a pleasure to be here to share with you some of the experiences that we have been having at IIT. I think it might be instructive if I were to develop this in sort of a story form, because as I look back on all the things that have happened, it is somewhat unbelievable even to me, and I think if you were to hear how it developed, it might appear to be a little more plausible.

The title of this Symposium is "The Uses of Computers in Education"; from our point of view, our concern is "Education in the Uses of Computers" so we turned it around a little bit. As Herb has pointed out, my primary function for a number of years has been as a member of the chemistry department in a private university--an institute of science and engineering--Illinois Institute of Technology in Chicago. The research in which I am involved has to do with so-called molecular physics--what we do is look at molecular systems as collections of nuclei and electrons, and we then apply some of the laws of physics. We ask questions about how the electrons move around the nuclei and how we can infer something about the bulk properties of these molecular systems from these first principle approaches.

Now applications of these laws of physics imply a considerable amount of applied mathematics and arithmetic and bookkeeping and so when a computer first really became practically accessible and available, it became quite important to those of us involved in this area of research. For all intents and purposes this happened about ten years ago when a machine which came to be widely used appeared, namely, the IBM 650. At that time, I introduced myself to the machine while visiting another university, Carnegie Tech, where a man named Allen Perlis was teaching.

Allan Perlis has had quite an impact on the world of computers, because he was the one who first created a practical general purpose translator. This was an emancipation kind of development because it meant that the kind of person who wanted to communicate with the computer was no longer constrained to write programs for the machine in the number code which the computer was capable of executing directly, but rather could write instructions for the machine using English and a form of algebra, and let the machine in one pass function as a translator, and in the second pass function as a computer. Allan Perlis, by the way, took his bachelor's degree in chemistry.

At IIT, the professors involved in doing graduate research and teaching graduate courses are also involved in the undergraduate program. In teaching junior physical chemistry, it became clear to me that we could with some profit introduce our undergraduates to the use of the computer in junior chemistry. We arranged for them to perform an experiment, then had them process some of their data by hand. This involved a form of curvefitting which turned out to be very tedious. The slide rule didn't give them enough significant figures, so they had to do it by hand--or, if they could find a desk calculator--do it that way.

Later on in the term we convened as a group and spent a six hour laboratory period on the computer. In one hour, I introduced them to the concept of a stored program computer, in the second hour introduced them to a set of eight instructions for a particular computer, and in a third hour went through a sample program with them. And these students were able to master those essential ideas in that short period of time. During the three hour period in the afternoon, we convened as a group and wrote a program for a least-square fit of a straight line for the data which they had collected earlier in the semester. The experiment was quite successful--the students demonstrated that with that short exposure they were able to learn the basic idea of a stored program machine and how to use it in the context of the discipline that they were studying.

We had all the positive features going for us--motivation, how the thing fit in, etc.--while our administration, since we are an engineering school, had been wrestling with the problem of: "OK the computer is here--we are an engineering school--it is obviously important to us--but, how do we fit it in?" I don't know how familiar you are with the undergraduate programs in engineering schools, but they are extremely tight and you can't talk about arbitrarily adding another three-credit hour course.

They had heard about what we were doing in chemistry, and asked me to come in and address the dean and the department chairmen of the college. I made a presentation and a recommendation, and they accepted the recommendation. I happened to have a post-doctorate student working with me at that time who was more interested in computers than in chemistry, so he went along and became the assistant director of the IIT Computation Center (there was no director). Well, that went along pretty well--they accepted the idea that it was not necessary to conceal the computer in the third floor of an obscure building on campus, but rather to put it in an open area, and to put printing keypunches all over the campus so that students could have ready access to the machine, and generally make access to the computer wide open. As a consequence, we had a large increase in the utilization of the machine. Now, this sort of put the professors in an awkward position, because the students were now coming into the classroom with sheets of computer print-out saying: "Gee, I did this, and I did that--and, can we talk about it?" So it wasn't too long before we found that these short non-credit courses we were giving for professors were becoming more and more populated.

Well, it was clear that there was need; it was also clear that there was an expense; and we understood at that time that the National Science Foundation was funding computer facilities--so I was asked to put together a proposal. The proposal included a budget item for a director. While the proposal was being considered, and after we had a site visit, we interviewed several people for the position of director--but when the grant came through there was no director. So in case you're wondering how a professor of chemistry gets to be director of a computation center--there was nobody standing behind me, so here I am!

The organization of the Computation Center might be of interest to you, because you might see your own institution in some way reflected herein. We have the same kinds of problems that a secondary school will have--but we are not supported by the taxpayer, and so we don't have mom and dad coming in scrutinizing what we are doing in infinite detail and trying to guide and advise us in infinite detail--but nevertheless we do have our problems. It was clear that the computer is going to be useful to us in three different ways--administration, research, and teaching.

The reason we have five departments in our Computation Center is because we made a fundamental policy decision at IIT along the following lines. We recognized that the modern data processing machine is indeed a general purpose device, but the power of the machine goes up roughly as the square of the cost. For example, our present installation costs us about $15,000 a month to lease. If we were to bring in the larger model from the same vendor which might end up costing us $30,000 a month to lease, the computing power available to us would not be just twice as great, but would be four times as great. This suggests that instead of having two or three independent data processing centers on our campus, we ought to coordinate our resources and have a single one. For the same expenditures of money, we would have much more computing power available. And this is what we have done.

Five departments comprise the Computation Center, and one is the Information Processing Center. That is where the hardware is. It has an operations manager who gets the work done, and it has a systems manager who maintains and upgrades the master programs which, together with the hardware, make up the utility. We also have an office manager who takes care of accounting and things of that nature.

Then we have an Information Science Center (which was discussed earlier today). What are universities doing in terms of reacting to the tremendous need for people trained in computers? The Information Science Center at IIT involves seven different courses--substantive courses--at the undergraduate level. One of these is a freshman course, and the reason we have this is because more and more incoming freshman already have learned to communicate with a computer using an algebraic compiler,and come to IIT expecting to have this kind of utility available to them. So we put up the one-semester-hour course for those students who are incoming freshman who are <u>deficient</u> in this regard. We regard this as a temporary deficiency, and we expect that the one-semester-hour course will no longer have to be given after the next four or five years.

The remaining courses, however, are at the junior, senior, and graduate level. The reason for this is that we want the student to get a good foundation in mathematics and logic and language before going into the more substantive courses. An undergraduate at IIT can major in English, or mathematics, or chemistry or whatever, and develop a very strong minor in what we might call computer science.

At last year's annual meeting in Washington of the Association for Computing Machinery (one of the two professional societies which has grown up around the computer), it was revealed that there are some one hundred universities which now give a bachelor's degree in what is called computer or information science. There are something like 25 universities which now offer a master's degrees, and some 15 which offer the PhD. For example, the University of Chicago has an MS and PhD program in mathematics; and an MS and PhD in applied mathematics; and an MS and PhD program in information science. I want to make that point because if I can only do one thing at this meeting, I want to dispel the notion that you equate mathematics and computers. They are not the same. Thus we have an Information Science Center--that is two of the five departments.

Then we have two departments which function as interfaces between the potential user community and the Information Processing Center. One of the buffer groups is called CASA (Computer Applications for School Administration). It is the purpose of that group to interface all the administrative offices of the school, the registrar, the bursar, the controller, etc., with the Information Processing Center. The purpose for channeling it this way is two-fold: one, to make maximum use of the expertise available to us, and two, to do this in a coordinated way so that we can move to a common data base for financial and student records. Three years ago at IIT, an incoming freshman was assigned three different ID numbers according to whether he was dealing with the registrar, the bursar, or the dean of students. Furthermore, three years ago, I don't believe our academic vice president knew exactly how many buildings we had on campus, let alone a comprehensive space inventory. This is the kind of thing that CASA is doing for the school.

The other buffer group is called CART (Computer Applications for Research and Teaching). We recognize that professors in our departments certainly know their own discipline and have come to be aware how their own discipline has been enhanced with the advent of the computer. We don't expect them to necessarily have deep technical knowledge of the applications programs and how to mount them on the machine and get them incorporated into the operating system, and so we have the CART group. CART applications programmers work with the professors in terms of developing special-purpose programs for them to assist in the classroom, as well as in research.

The fifth department is our secondary school program, which is the main reason I am here. And this now has the status of a separate department, although the only full-time person associated with it is myself--and in an increasingly remote fashion. I mentioned that my involvement with the computer game at IIT came via our junior physical chemistry course. I was invited to address a group of high school students at Oak Park's River Forest High School. I happened to live in Oak Park, and my nephew happended to be a member of the club, and he wanted me to come out and give them the usual 40 or 50 minutes of general description. Instead of doing that, I offered to repeat for them what I had been doing routinely for juniors in physical chemistry. I found that with one and a half hour meetings on each of three successive Saturdays, they were able to grasp the basic ideas. I then brought them to our campus and attempted a run on our UNIVAC 1105 (which goes back two generations in computers, although it only goes back six years in time). I found that these students were highly motivated and well able to master the basic ideas, so I suggested to our administration that, since we had unused capacity in our machine, we make this kind of an opportunity available to a larger group. So we sent letters out to a couple of hundred high schools; I had planned a program based on 150 students for Saturdays on our campus. Within a couple of weeks we had 700 applications, and these were all top drawer students. We doubled our program, and the man I had retrained for this purpose offered the same lecture two Saturdays in a row, so that he could accomodate 300 students, 150 at a time.

Well, from that simple beginning the program has come to the point where there are eight different courses offered on Saturdays for high school students. These students are grouped by three levels, but I won't go into this in detail. This information is given in the article which Mr. Peluso has written, and which will be published in summary form in "American Education" in the May issue. (It is included in full in the Appendix.)

In addition, we found that after the students had started through these courses, we began to get some feedback that the teachers wanted something. And so we put together workshops for the teachers. The introductory workshop is now being offered for the twenty-fifth time over the past five years. The cumulative total of high school students who have come through the Saturday program--at least through the first course--is 10,000. The cumulative total of teachers who have come through the Saturday program is over 1,000. The number of high schools represented within a 100 mile radius of Chicago is something like 380. In addition, over the last couple of years, junior high schools have come to be involved as well. In the Chicago public school system alone there are fifty elementary schools who have had either teachers or pupils come into the student program or the teacher program.

This program has received no Federal or state support but is supported by modest participant fees. Because there are large numbers of participants involved, the fees are sufficient to cover the stipends of their instructors. There are nineteen instructors involved in our secondary school program; five of these are computer professionals who come out on Saturdays to give the more advanced courses. We have drawn on the professional computer community for this support. Two of the remaining instructors are elementary school teachers who have just had a book published by Addison-Wesley called "Basic Fortran," a self-instructional text that is a follow-up on the self-instructional programmed text on IITRAN/360 (the language we created ourselves) which was written by two high school teachers and the creator of IITRAN and is also published by Addison-Wesley. Twelve of the instructors are high school teachers--I think eight of them are from the Chicago Public School System. Now what we have had happen is a kind of phenomenon where people in the university and teachers from the secondary school system have come together to address themselves to the basic problem: The computer is here, what can we do about it? The computer vendor, at least up until the recent past, has never really addressed himself to the problem of supporting mass education with the computer. His primary concern has been with the fact that the computer is used widely in industry. About 80% of the computing which is done is in data processing, and approximately 20% is in scientific and engineering work. Only recently has the potential of the educational market made itself apparent to the vendor.

At IIT we are sort of under the gun as far as funds are concerned and so we have to make the best use of what it is that we have available. So after examining the various languages we had available for the equipment that we had, we found that none of these really suited our purpose. We created our own language (IITRAN) and designed our own supporting compiler, designed to facilitate student use of the computer. To give you some idea of student program demands in terms of computer time: we take the monthly rental, double it (the additional 100% being overhead, staff, and so on), and divide that by 176 hours (which is the number of hours in a prime shift during the month) to arrive at our rate -- on that basis and at that rate, we can process a typical secondary school student problem for about 25¢. I think that makes clear that the cost of providing computer support to the high school student can no longer be considered a barrier to using the computer to support mass education.

Our first teacher workshop involved six Saturdays (a teacher by the way gets no credit for this, except as the local school board may want to permit this as credit towards advancement towards the next pay scale; the teacher puts in his own time; and the teacher pays a fee out of his or her own pocket) and, after six Saturdays in the workshop, the teachers are communicating easily with the computer, using one of the languages which is available.

Now I want to make clear something which I found to be a point of confusion and misunderstanding as far as school administrators are concerned. Our program is not concerned with the use of the computer to support the administration of the school. We are not concerned with computer-aided instruction, except insofar as a clever teacher might be able to relate the student's almost romantic involvement with the computer to whatever subject the teacher is teaching, and use that as a "carrot" to draw the student along (but, in a technical sense, this is not computer-aided instruction).

Also, this is not vocational training -- we are not preparing students to be programmers, to be board wirers, to be computer operators. The trade schools are doing an adequate job of this, and besides, at the rate at which technology is advancing, these will be short-lived trades in any event.

What we are concerning ourselves with is what one of our previous speakers showed when he outlined four different disciplines -- computer science, physics, some aspects of electronics, and mathematics. We are concerning ourselves with something that is called computer science, and this has to do with computer systems and sophisticated computer applications.

Before I leave this question of the universities and the academic discipline which is emerging, those of you who are involved in trying to define what you mean by computer science may wish to make a notation somewhere at this point. The professional society called the Association for Computing Machinery (ACM) has behaved in a very responsible fashion. It recognized very early in the game that one of the biggest problems that our society faces is that it doesn't have enough people educated adequately as to what these devices are and what they can do. And so the ACM selected (and got on a voluntary basis) a number of people who are doing pioneering work at major universities to form a committee to try and define what this emerging new discipline is, and furthermore, to define courses which collectively could be the basis for an academic program in computer science. The entire issue of the March 1968 ACM Communications, will be devoted to a definition of the emerging discipline, computer science;

comments about its interface with other so-called disciplines; a collection ofcourses (with a fairly detailed description of each of these courses); and a highly selective bibliography. Those of you concerned with addressing yourselves to that problem will probably find that to be a useful reference.

Now we make the computer available to the student in order that he can program solutions to problems. You notice that the computer doesn't solve problems for you--all it does is do the clerical and arithmetic work associated with the solution. In addition we are also trying to get the students and their teachers to think about certain computer applications programs where one doesn't actually write a program for working out a solution to a problem, but rather somebody else has worked out a general-purpose program and written it in such a general way that a whole class of problems can be accomodated with it. For example, we happened to have IBM equipment, and one of the general-purpose applications programs available to us is something called the General Purpose Systems Simulator (GPSS). This is a segment of our environment. This can be a complete abstraction dealing with the thought process or something as mundane as regulating traffic through a supermarket--the check-out situation.

The important point here is that we want to elevate the students' and teachers' sights. We want to make it clear to them that the so-called computer--the word, by the way, is a misnomer--is really an information processing machine. The most common kind of symbols with which we are acquainted are the numbers and their manipulation, so the computer gets to be used as a giant calculator. But let's not mislead ourselves that this is going to be the most important application to which these devices can be put.

We want to elevate the students' and teachers' sights to think about simulation--model building on the computer. In our view, this is one of the most important areas to which we can address ourselves. Our high school courses, by the way, actually get into the use of GPSS-III; they get into linear programming; they get into matrix techniques, and so on. Obviously, only a small percentage of the students after the first course will go into those very advanced courses, but I mentioned that just to give you some sort of feeling for the breadth and range of what it is that we are concerning ourselves with.

As a sidelight I would like to mention something else; most of you are people whose profession is education with a capital "E." In a sense this is not my forte, but I am dealing with people who are part of educational systems. It has been a very interesting and rewarding experience which I don't seem to see replicated elsewhere; I am able to work very effectively with high school teachers on a peer basis. In a condition of mutual respect, we address ourselves to the basic problem; we sit around the table and kick around what it is we want to accomplish, and how we are going to accomplish it. I think to the extent that we have accomplished something--and I think we have--this has been a rewarding process. I hope that by the way I am describing it I don't over-emphasize my own role in this. I got the program started and help to guide it, but the program is being carried on by the people who comprise the secondary school computer science education program.

The title for my presentation here today had the word "Compu/Tel" in it, and I haven't gotten to Compu/Tel yet. Our present state of technology in terms of the physics of the components and the electrical engineering of the components, electronics, is such that people who assemble these devices that we call information processing machines are probably only utilizing about 10% of the potential available to them. These hardware devices, moreover, are not very useful to us without the associated master programs (software), executive routines, systems development, and the like. And I would guess that the systems people are probably utilizing something like 10% of the potential available to them in hardware--hardware which is only realizing about 10% of the potential of the technology which is supporting it.

The average user who writes programs for the machine is himself only using about 10% of the capability of the software, which is using only 10% of the capability of the assembled hardware, which is only using--well you see what I am getting at! We are so far behind in terms of awareness of what is available to us that it is pathetic. Even if the development of computers were to stop today, we would still have a lot of slack to take up in terms of just what these devices are, what they may mean to us, and what we can do with them now that they are available.

I mentioned earlier that we at IIT had concerned ourselves about the cost of providing computer support to mass education. And we became aware that the superintendent of a school district is under tremendous pressure from his community--everybody has heard about computers, there are a lot of enthusiasts who run around saying all the things that should be done with them and so on--and so the superintendent really doesn't know what to do. On the other hand, he is always under the gun--he has to produce performance for dollars spent, his budget has to be very tighly laid out, and he is continually being monitored by this civic group and that civic group. And yet he knows via the grape vine that getting into the computer business is like an iceberg: you only see the one-ninth that is above the surface, you don't see the eight-ninths below the surface. He hates to put himself in a spot where he is going to sign a blank check. He doesn't know what he is going to be getting into as far as hardware, rental, software support, additional people are concerned, until he is really committed--you can't tell whether or not a man is going to perform even though he says he will, and so on. So we attempted to do something for the school superintendents in the Greater Chicago area.

We put together a package. We said to the superintendent: if you are willing to spend $2300 for an academic year, this is what you will be able to do. You go to Illinois Bell and have the representative come out and install a teletypewriter. Now a teletypewriter is an electric typewriter with two little boxes on the left hand side, one of which produces punched paper tape, and the other which reads punched paper tape. If you want the technical definition, it is called an ASR 33, ASR for Automatic Send-Receive. Herb has one over at CEMREL, if you want to see it. They charge $25 to install it or something like that. And the total cost will be about $100 a month to rent a teletypewriter, the Dataphone which interfaces the teletypewriter with phone system, and one business line. For a 10-month academic year, there is $1,000 plus installation cost, which accounts for $1100 of the $2300.

Then we told him we will install an interface between our computer and the Bell Telephone System to support you (you collectively being the superintendents). On the basis of how many of you decide to come in, we will make up a pro-rata share of the cost which we will assign to you. The pro-rata share turned out to be $50 a month. This was for a piece of IBM equipment, and also for some Illinois Bell Dataphones. That comes to $500 for the 10-month academic year; we are now up to $1600. Notice nothing has happened as far as IIT is concerned--and I got some awful glances from our business officer! Well, this left us with about $600 to work with. The $600 we converted into CPU time. Now CPU stands for Central Processing Unit, and for those of you who are going to get into the game, if you haven't learned what the term means you want to distinguish between having control of the equipment by the wall clock, versus being charged for the time the machine is actually working on your program, and by CPU time I mean the latter.

With the software development that we had introduced--namely the development of the IITRAN language and the supporting compiler--we found that we could process student problem at the rate of about two seconds a student problem (that is for compilation and execution).

Let us translate that into operational terms. An hour has 3600 seconds, so two seconds per problem means that one hour of CPU time will accomodate 1800 student problems. So with that $600 for CPU time we found that the high school teacher in the classroom could support a class of 50 students, each student submitting three programs per week for 40 weeks, which comes to 6000 student programs.

Well, a number of the schools were quite interested in this, and they took up the offer. And this gave them an opportunity to learn first hand what the computer is and what it can do for them in the context of a computer-science situation. They also learned some of the jargon. Earlier, a speaker made reference to the fact that people in the computer game have built up their own kind of language. I don't think it is done intentionally to isolate themselves, but nevertheless there is a jargon, and this jargon tends to frighten people off. Actually the computer is an extremely simple device, and the vendors are working even harder to make them even simpler. Let's face it--the vendor is in business to make money, and he is not going to make something which no one can use. He is going to work very hard at making it easier and easier to use.

We have implemented this past week something called a remote job entry system. That is a technical term, but I do want to dwell on it a bit, to throw that in contrast to so-called conversational systems. Usually by a conversational system, what is meant is the following: A person can sit at a keyboard device (or, if he is very well off, a cathode-ray-tube device with a light pencil), and communicate directly with the computer, and to all intents and purposes, the responses to the man sitting at the keyboard are instantaneous. The man sits at the keyboard; he types in a line of code and transmits it to the computer. That line of code is assimilated. It is either accepted, or back comes some diagnostic message telling him that he misused the language, and to make a correction. And then the man goes on to the next line of code, and so on. Roughly speaking this is what we mean by conversational.

Remote job entry is rather different. Remote job entry means that a person sitting at any one of a large number of consoles which may be connected to the computer, is in effect sitting at the principal console of the computer. That is, he sends in a job and it gets stacked up in a line of jobs waiting to be performed by a particular applications program, and at the time that the scheduler in the machine determines by priority that it is time for that line of jobs to be processed, the major translator or applications program (for example, it might be GPSS) will be loaded by the machine and then the stack of GPSS programs will be processed and the results will go back into some form of intermediate memory and wait there. Then the user from the remote terminal can "tickle" the system at some later time and get his output. So he is batch processing.

Now the difference between these two approaches, as currently implemented, is that most systems currently available to support conversational mode are quite expensive in terms of cost per calculation, because the overhead associated with doing all the bookkeeping (keeping track of everybody's line of code, etc.) is very large. Furthermore, most common conversational systems are dedicated systems--that is, they support just that one language in a practicable way. A remote job entry system, on the other hand, permits a general purpose computing machine to have many compilers and applications programs on tap, as though it were a library, and anyone at any one of the remote terminals which has access to that particular system can use any of the compilers or applications programs. So you have to have clear in your own mind the distinction between having conversational access to a particular programming language, which enables you to program solutions to a problem, and being able to have access to many programming languages, and to applications programs (where you do not do any programming, but where all you have to do is learn how to formula e the input parameters, and how to read the results when they come out). That is quite a distinction.

Anyway, we support remote terminal users via the remote job entry route and remote job entry means that anyone who is hooked into this system can send a job in <u>at any time</u>, and can tickle the system <u>at any time</u> to see if the job is indeed ready. Incidentally, we have all but eliminated the operator's job, because the remote job entry system has its own scheduler.

Do you know that old biological expression: "ontogeny recapitulates phylogeny"? Well, we are making an opportunity available to educators to start in on the ground floor and start learning about computers the way the rest of us who are in the computer game learned about computers--and that is, to actually start using computers. I think that we are learning about the tremendous needs for education--not only about how to use the computer, but about its implications and its impact on society. We started a new course at IIT just last week--you might be interested to know that I have two instructors involved: one of them is a systems analyst who works for United Aircraft as a computer programmer, and the other one is a graduate student in the divinity school at the University of Chicago working part-time as a systems programmer. The title of the course is "Computers and Society." The reason we started this course is because it is our feeling that these devices are becoming so widespread--it is estimated that there are some 45,000 of them operating now in the U.S., and that the number will double within the next six years--that they impact every area of human endeavor. I can't conceive of a single area of human endeavor where the computer couldn't be used.

And yet here is another example of a technology where in terms of basic human values we are not making basic decisions. I don't know whether or not you noticed this in last week's Wall Street Journal. There was an article on credit bureaus, and I was flabbergasted to discover that there is an association of 2200 credit bureaus which has individual files on over 100 million people, and very good cross-communication. Any man who has a business can buy a service from these credit bureaus, and get a run down on anyone. Almost anyone can have access to a person's file--except the person himself. You have no control over it, you have no way of challenging the veracity of the information stored there about you, and yet this can have profound implications--the FBI gets into this file, the State Department gets into this file. Last November, President Johnson created a Public Broadcasting System, and had very high level people elected to the Board of Directors; he started out with the President of MIT, and Milton Eisenhower. Now, he is talking about creating an information bank. Yet how can society as a whole begin to react to this, to let its wants and needs be known? The need for education is not only in the use of these computers, but also for education about their tremendous impact. Meeting this need is in the hands of the establishment--the education establishment. Well, I hope that through conferences such as this one--each of you is an influential people in your own school systems--we can begin to come alive to the problem that is before us and that we can begin to think constructively about how we might begin to face it. We are witnessing the marriage of the communications industry and the computer industry, and this whole thing is going to be national long before you realize it.

You might be interested in one more reference: there is an organization of about 70 universities around the country called EDUCOM. In the summer of 1966, 180 or so scientists met at the University of Colorado and they gave very serious consideration to the notion of creating a university network that would span the entire country, coupling computers. They looked at all the technical problems associated with this--the fact that different computers use different languages, the interface problems of the computer with the different communications systems, the problem of security of files, etc. A report of the conference was published by Wiley in June, 1967. The title of the report is "EDUNET". For those of you who might find yourself getting involved in a network, or some subpart of a network, I think you might find that that would be a very useful reference document. In layman's terms, some of the basic problems are stated there, and you want to at least be able to ask the right questions even if you don't know all the answers.

Well I didn't mean to convert the podium into a pulpit, but I couldn't help but express a few views about how things are being done, and how I hope things might be done. For those of you who are going to get into the time-sharing business, there is an article that appeared in Fortune in August of 1967, called "Computer Time-Sharing--Everyman at the Console." You might take a little comfort from some information which is revealed there--all of us feel terribly ill at ease because there is so much to know about computer technology, and we know so little of it. We are terribly afraid of stubbing our toe or making a mistake, and sometimes it inhibits us from going forward. You might draw a little comfort from the fact that two major companies have stubbed their toes very seriously. They were going to compete with each other in creating the super computer, which was going to be super time-sharing, which would have all the best features of conversational and remote job entry. IBM entered this field with the 360 Model 67, and GE entered the field with the GE 645. Both of these are referred to in the article. Here we have a case within the computer industry itself, where you think they would know better, where internal enthusiasts let their enthusiasm carry them and their companies down the primrose path.

Well, I hope that these remarks have been stimulating, and that some of the information conveyed to you may be useful. I would certainly be happy to amplify on any of this either here or at some later time. And I would like to thank you for this opportunity to speak to you.

DISCUSSION: PETER LYKOS

Question: I'd like to hear a few more comments on model building.

Reply: Model building can be classified into two forms. One, where one creates a mathematical model where everything is determined, such as a collection of equations which might represent a satellite in orbit, and the basic equations can be parameterized in such a way that by changing some of the basic parameters you might change the eccentricity of an orbit, or even go to limiting cases where the satellite might plunge into the earth or actually leave its orbit and leave us indefinitely. Now, that would be a simple kind of model.

As we get more sophisticated we can talk about how we could introduce the effect of the moon and other planets on position of the satellite, and we could even talk about a random number generator which would assign uncertainty to the data which we are ostensibly getting, so that we could make the model even more and more realistic. That would be a more or less completely deterministic type model.

Another type of model has to do with the traffic situation, such as the flow of traffic in and out of an airfield or in a supermarket; in this case, the particular application program I made reference to--GPSS III--is a general purpose system simulator. A supermarket model is a model system that has to do with a traffic kind of situation. Now we can't tell a priori how many customers are going to be coming through the supermarket at any time but we might know over a period of time the total number of customers that might be, and then we use a random number generator so we have customers coming at random in such a way that the sum total over a given period of that will be the actual total number.

There is also something which is called "queueing" theory. It's kind of a funny word spelled in a particular fashion, which has to do with the theory of how pieces of material, or people, or whatever, line up in channels which are available to them and through which they must pass to go from position A to position B. There is some fairly sophisticated theory involved, which is the basics of GPSS III. Now, there are books which are available dealing with this kind of thing. This is part of a subject which is sometimes referred to as operations research. If you talk to the people in the local university who are in industrial administration or operations research, I think they could refer you to books or semipopular articles.

Ohlman: There is an organization devoted to this art--Similation Councils, Inc.--and they publish a journal, but it deals mostly with analog computer applications, I believe.

Lykos: Also, you will find that in some of the major universities where they have sufficiently large computers, they will talk about simulating some segment of our economy or simulating the operation of a bank. What you can do is take all the information from the stock market over a period of time and make this available and have a man act as though he were vice president of the bank in charge of investments. And since what he does to the stock market isn't going to perturb the stock market very much, he can make decisions and guesses, but instead of waiting for time to pass on the ordinary scale, he can have a period of years pass in a very short period of time. So that a man who is playing at being the vice president of a bank in charge of investment can have five years of experience in a week. So there you've simulated the operation of a bank, and simulated a segment of our economy.

As a professor who deals with students in a classroom, I see this from another point of view. I can say, here I am, and I think I understand, say, the investment business, and I create a model, and I put into that model all the things which I think are important, and make guesses as to the values of the critical parameters. And then I turn the students loose on it. Of course, the sharp students find loopholes, and suddenly we have a few "paper" multimillionaires,--but then these loopholes become apparent to me and point out to me my own imperfections in creating that model, and so I then correct the model--and it is by this feedback process that these models can be made more and more precise. However, we must never lose sight of the fact that we are <u>simulating</u>, and this is in itself <u>not objective reality</u>. This idea of model building does force one to think abstractly about some segment of the physical or natural environment, and to try and find some order--and it's a test.

<u>Ohlman</u>: In the ultimate, you get into a situation which is actually a game, as in "operational gaming" developed by the military, where they try to give people--the people they're trying to train--a situation so real that they believe they're actually in the situation. In fact, I believe that in the SAGE system of air defense, they actually have two computers back to back. One of them is running the real data, and one of them is running the siumlated data, and the people operating the system can't tell which is which.

<u>Question</u>: Dr. Lykos, do you find that the teachers and students that use the terminals are satisfied with it in terms of scheduling, and do you find it necessary to restrict the length of the program or the amount of outputs?

<u>Reply</u>: I'm glad you brought that up. One of the features of the IITRAN/360 System is that the teacher has complete control by setting certain parameters. The teacher can limit how much computer time each student will use, how many lines of output, how many lines of code will be processed, and so on. The need for this is quite apparent, because one imaginative student can shoot your budget in one afternoon.

The whole program has evolved out of a cooperative venture between the technical people of the university associated with the computation center and the teachers who are actually using the system. All these little things which crop up--and I couldn't begin to make a list of them--have had attention given to them and have been incorporated into this teacher-student oriented total system.

In terms of waiting for results, this is a psychological thing. Once you are used to cake, I guess bread tastes pretty dry. And if you are accustomed to very sophisticated equipment, where you have a tremendous amount of computing power, and a rich variety of languages, and immediate access, it becomes a little difficult to be satisfied with anything less. On the other hand, if you have had nothing, and you begin to work your way into a system through successively more sophisticated levels, there is really no reason for dissatisfaction.

Certainly when you are sharing a large system, you will have shortcomings. The teletypewriter is a fairly slow device compared with some printers. An earlier speaker made reference to the fact that a new high-speed printer shoots paper through so fast that you can see it flying through the air. We might be amused, but such high speed printers are even more dangerous than that. When you have that sheet of paper moving through so rapidly, you have a van de Graff generator. If that high-speed printer isn't appropriately grounded, you can be seriously hurt when you come up on it because it generates quite a potential relative to ground. Now that is extremely fast compared with a teletypewriter--which types at a rate a very fast typist would type.

On the other hand, there is another problem: These devices which are so fast can produce a tremendous amount of output, but you have to now learn some kind of sophistication in asking the right questions. Sure, I can always ask for anything that the machine is capable of giving me about the problem that I solve, and I may find myself with reams and reams of paper, tables and tables of numbers, and then have to ferret my way through this to find out what it is all about.

I have a friend at the University of California at Berkeley, and he tells me that the University buys paper that goes through a high-speed printer by the boxcar load. And when the computer center calls you up to tell you your program is ready, you are told how much output you have by linear feet of fan-folded paper. Now, I don't know how long it takes those users to get useful information out of that. This does force the student to be more selective in terms of what he's asking for. But, you have really asked a complicated question, and I don't think I have given you a complete answer.

Question: We've heard a lot about IITRAN, but I wonder if you can tell us the effective compiler speed for student-type programs, in terms of instructions per second?

Reply: I brought with me, at Herb's request, five packets which have a lot of material relating to this, but I don't have it at my fingertips. I could give you some technical information, like: it is an in-core compiler; it occupies, I think, something like 37,000 bytes. In the package, we have copies of a self-instructional text with worked out examples including actual print-out where the computation time and execution time is given for a variety of problems. So rather than give you some offhand answers, I would have to refer to it in that fashion. If you like, I will give you the telephone number and name of our system's programmer who puts this together, and he will give you a lot of detailed technical information about it. But it is very competitive with other in-core compilers.

Even though one can fill the entire Hollerith card (not just one statement), IITRAN program processing is limited by the speed of the 1,000-card-per-minute card reader. The thing which distinguishes the language in the first place is that it is relatively clear of the arbitrary restrictions which are part of such languages as FORTRAN.

It is a very free-flowing language. It was designed by a man who was on the PL/1 Committee for three years and who worked closely with our high school program. It also has very comprehensive diagnostics. It is a semi-forgiving compiler. For certain errors--which are common ones--the compiler will assume what you meant and then will go on and process the program as though you had in fact not made an error, and let you know that it made that correction.

What we find is that the program is so fast that detailed accounting of the student's use of the machine may be as much as eight times the cost of his actual use of the machine. And so, for a while, we were selling ID cards for 25¢ each, which would allow the student to have up to so many seconds of machine time without any identification as to who the student might be or what he was using it for. Now there is something for you to think about.

Question: Do you want to say a few words about the availability of this system?

Reply: This kind of information is given in great detail in Mr. Peluso's article. There are several appendices which give explicit information, such as: "What are the features of the IITRAN language itself?," and "What are the equipment requirements for supporting IITRAN/360?" Mr. Peluso says that a school could ride piggy-back on somebody else's IBM 360, the idea being that very few schools are going to be able to afford a comprehensive large computer system for themselves, so it behooves them to have convenient access to such a machine.

IITRAN/360 can be loaded just as with any other program, such as PL/1, Fortran, Cobol, Snobol--you name it. Incidentally, at last count, there were almost three hundred computer languages--so don't think the world is married to any particular one. Furthermore, it was pointed out about six months ago that there were twenty mutually incompatible dialects of Fortran IV, so the fact of talking about yet another language shouldn't bother you. IITRAN/360 can be loaded into an existing system, and one can have card input and high-speed printer output, just like any other program designed to be interfaced with DOS--and it is now being implemented under OS at Cornell.

In addition, it can also be accessed from a remote teletypewriter through an IBM 2702 interface (which is the IBM interface to the Bell System). And this information is all given, in sufficiently detailed form, that you could show this to someone who is an expert on computers, and they could tell you from this whether or not you could support the system.

The remote job entry system which we have is also available free of charge to academic institutions, and that enables you to use PL/1, Fortran, Cobol, IITRAN, Assembler Language, STOP, or whatever is in the systems library, from the remote terminal.

Ohlman: I don't want to leave anybody with the impression that having 300 different computer languages is really quite as bad as having 300 human languages. This is because all computer languages have to follow, more or less, the same rules. In fact, there have been a number of attempts to actually develop programs to translate from one computer language to another, and these have had a lot more success than translating from one human language to another.

APPENDIX: A REPORT ON AN ESTABLISHED REGIONAL PROGRAM OF COMPUTER SUPPORT FOR SECONDARY SCHOOL COMPUTER EDUCATION

by Anthony P. Peluso, Chairman, Mathematics
Lane Technical High School, Chicago

Day by day, computers play an increasingly important role in our lives. The present high school student will live and work in a complex, highly computerized society. Presently, most high schools have no program of computer education. The secondary school curriculum cannot continue to ignore this area of education for very long. In the near future, if not at the present, high schools which lack some degree of computer activity will be considered out-dated. Administrators of secondary schools and colleges, in attempting to inaugurate a program of computer education, face several formidable obstacles. Computer equipment, so necessary to provide a realistic experience, is very expensive. Teachers with computer training are scarce. Standards for course content and curriculum organization in relation to computer education are not well defined. Apprehension, mystery, and, in some cases, fear are associated with the computer.

A few schools have overcome these barriers and have managed to purchase or lease a computer. Although usually not of recent vintage, and limited in size and speed, these devices can process such routine clerical tasks of the school as payroll and report cards, thereby justifying part of the cost. When the computer is not being used for record keeping, it can be made available for educational purposes. This arrangement, although somewhat workable in schools which find it economically feasible, has certain undesirable restrictions. Since the machine is limited in capacity and speed, only a small number of students can be accommodated. Furthermore, the size of the machine limits the types of computer activities the students can experience. Few, if any, high schools can afford to purchase or lease a modern computer, and it is unlikely that they will be able to do so in the near future. Fortunately, this does not mean that high school and college students cannot be provided with the valuable experience of interacting with such a device. For several years, commercial, educational, and federal institutions have been concerned with this problem. Many experimental programs in computer education have been in progress for quite some time. The future will certainly see the computer used in education in many diverse ways. One area of serious concern is the problem of providing computer education to large numbers of schools and larger numbers of students at a reasonable cost. Results of experimentation indicate that one answer lies in the regional approach.

Due to recent technological developments, it is now possible for as many as fifty or more schools to share a central computer facility. Remote terminals installed in the schools can be linked by telephone lines to a modern computer which has time-sharing capabilities. Until recently, this was considered a futuristic dream. It is now a reality, and the expense can be put well within a school's economic reach.

Last September, by starting its second year, a program of computer science education called Operation COMPU/TEL demonstrated the feasibility of the regional approach in the greater Chicago area. All equipment required for this regional plan is standard and currently available, making it possible to duplicate the entire plan of computer support for secondary schools and colleges in other regions. The details are presented here.

OPERATION COMPU/TEL

Illinois Institute of Technology, in conjunction with Illinois Bell Telephone Company, has brought the computer directly into the high school classroom via a telephone connection with the IIT Computation Center. This exciting new program, which began in October 1966, involves public and parochial schools from Chicago, its suburbs, Gary, and other communities in northwestern Indiana.

Each of the participating high schools has installed an Illinois Bell 33ASR (automatic send-receive) teletypewriter, an inexpensive, reliable, compact unit about the size of a telephone stand. The teletypewriter, when equipped with a telephone line, may be used for transmitting a problem directly to an IBM 360/40 computer at the IIT Computation Center. In a short time, the results come back from the computer, with the teletypewriter rolling out a typed copy.

Far-sighted administrators of participating schools have provided their students with a valuable modern educational tool which can be used in such varied fields as mathematics, physics, chemistry, biology, and business. In some instances, the school's computer program has been designed to include adult education courses.

Each participating school is free to design and develop its own computer science program. Since facilities, curriculum, available trained personnel, and other resources vary from school to school, several different approaches to computer science education have emerged. In some schools teachers have sufficient training to offer a one-semester or even a two-semester course in computer science. Other schools have revised existing courses to include a unit on computer programming. Most schools began their program by starting a computer club activity. In a few schools all of these approaches to computer science education are in practice.

With Operation COMPU/TEL well into its second year, some of the effects of providing students and teachers with the opportunity to directly interact with a modern high-speed computer are quite evident. Fear and superstition gradually diminish and are replaced by a realistic appraisal of both the capabilities and limitations of this powerful new device. Students begin to think in terms of solving the problem rather than solving a problem. Science Fair projects in mathematics and science that were previously avoided because of numerous required calculations now become possible. In the Chicago Public Schools students may now compete in a recently introduced Science Fair category -- computer science. Students are forced to think more logically and exercise care in preparing instructions, since faulty logic and careless commands are dealt with very harshly by the computer. Students communicate more with each other when working with computers; a spirit of camaraderie develops among users. Soon after exposure to this new field, students ask questions such as: Which colleges or universities offer degree programs in computer science? What are the requirements for part-time jobs? What career opportunities exist in this field?

Participating Schools

Operation COMPU/TEL was launched with fifteen secondary schools. The second year of this new program began with twenty-five schools, including two Chicago city colleges. Currently, the participating schools are:

NAME OF SCHOOL	CITY	STATE
Lane Technical High School	Chicago	Illinois
South Shore High School	Chicago	Illinois
Chicago City College Wright Branch	Chicago	Illinois
Chicago City College Southeast Branch	Chicago	Illinois
Chicago College of Osteopathy	Chicago	Illinois
Addison Trail High School	Addison	Illinois
Downers Grove High School	Downers Grove	Illinois
York Township High School	Elmhurst	Illinois
Larkin High School	Elgin	Illinois
North Junior High School	Dekalb	Illinois
Oak Park-River Forest High School	Oak Park	Illinois
Willowbrook Community High School	Villa Park	Illinois
Leyden East High School	Franklin Park	Illinois
Leyden West High School	Northlake	Illinois
Highland High School	Highland	Indiana
Munster High School	Munster	Indiana
Lowell High School	Lowell	Indiana
Lake Central High School	St. John	Indiana
Frobel High School	Gary	Indiana
Andrean High School	Gary	Indiana
Edison High School	Gary	Indiana
Emerson High School	Gary	Indiana
Horace Mann High School	Gary	Indiana
Roosevelt High School	Gary	Indiana
Tolleston High School	Gary	Indiana
Wallace High School	Gary	Indiana
Wirt High School	Gary	Indiana

The city schools of Gary, Indiana, obtained sufficient federal funds under Title III ESEA to support its participation in Operation COMPU/TEL for a two year period.

Estimated Expenses

As an example of the minimum expense involved in establishing instant communication between a high school and a major computer center such as that at IIT, the use of an Illinois Bell teletype unit costs Chicago-area schools approximately $150 per month for each installation. Processing a simple student problem costs about ten cents in computer time, a much smaller sum than the cost of circulating a book from a large city library. In a school with a Chicago telephone exchange, a reasonable estimate of total equipment and computer time cost to support twenty-five to fifty students for one academic year is $2,300--a low cost not elsewhere duplicated.

To accommodate a greater number of students, any number of additional offline (not connected to phone line) model-33 teletypewriters may be rented for approximately $50 each per month, or purchased outright for about $700. These units permit students to prepare their programs in the form of paper tapes for transmission via the main on-line send-receive teletypewriter.

An important factor in keeping costs low is the use of IITRAN/360, which enables computers to process relatively simple problems, in quantity, much faster than the older computer languages designed primarily for use in advanced scientific research or for commercial or industrial purposes. The great reduction in artificial language restrictions and the very comprehensive error diagnostic capability enhances the effectiveness of the new language which, together with the improved efficiency of large advanced computers, makes feasible the routine use of advanced computers by secondary school students.

IITRAN/360 is currently being used at IIT to support student classwork in several courses offered in various departments. IITRAN/360 was introduced at the University of New South Wales, Kensington, Australia. Approximately 1000 students learned to program in IITRAN, and the New South Wales installation reported processing more than 300 IITRAN/360 programs per day, averaging two seconds per program on an IBM 360/50 computer.

IITRAN/360, like other computer languages,is essentially a means of translating a few complex statements in English into an equivalent set of a great number of simple instructions which are comprehensible to a computer. IITRAN/360 can be used by students with no background in computer programming after about one hour of instruction.*

To facilitate the learning of IITRAN/360, a textbook titled IITRAN/360 Self-Instructional Manual and Text by Charles R. Bauer, Anthony P. Peluso, and William S. Worley, Jr., has been written and is available through Addison-Wesley Publishing Company, Reading, Massachusetts.

SCHOOLS RIDE "PIGGY-BACK"

The IITRAN/360 teleprocessing system was designed to support student use of a computer. Student programs are processed on a standard IBM 360 computer under the control of a standard IBM 360 monitor system, which gathers and stores student programs from the remote terminals while other jobs are being run. These developments make it possible for schools to use IITRAN/360 without seriously interrupting the normal job flow of the computer to which they are connected.

---

* The main features of the IITRAN/360 language are given in Appendix A.

Because of these technological advances, it is now not only possible but feasible for a school system to make an arrangement with an existing IBM 360* installation, be it in a bank, oil company, department store, or an educational institution, to append an IBM 2702 interface between the 360 computer and the telephone system. This equipment and ordinary telephone lines provide the necessary connecting link between the computer and the remote teletypewriters. Student programs can then be processed in "piggy-back" fashion, without unduly disturbing the operation of the 360 installation. Thus, the IITRAN/360 teleprocessing system can provide student access to a powerful, modern computer without the often prohibitive cost of installing and maintaining a computer center at the school.

In February, 1967, the U.S. Government Printing Office issued a publication entitled: "Computers in Higher Education -- Report of the President's Science Advisory Committee." The following excerpts are from Section V, The Computer and Secondary Education:

> "Computing is best used in secondary schools by means of convenient facilities, such as remote consoles, and simple instructional programming languages. Instruction in the nature of computers and computing can be by means of special texts supplemented with specially designed experimental equipment.
>
> "Unfortunately, this approach is contrary to much that is now being done in secondary education. Sometimes the computer used is one which is used for administrative purposes, which may be ill adapted to proper introductory instruction. Sometimes the computer used is a small machine purchased or rented primarily for instruction, but awkward to use and of limited computing power compared with a remote console attached to a large modern machine -- or even compared to job shop operation by courier or mail on some accessible more powerful machine.
>
> "We urge that the Office of Education and the NSF jointly establish a group which is competent to investigate the use of computers in secondary schools and to give schools access to past and present experience. Cooperation between secondary schools and universities, and particularly providing service to secondary schools from university centers, should be encouraged.
>
> "Computers and computing are already invading junior high schools and elementary schools, and this same recommendation should be applied to junior high schools and elementary education."

---

* See Appendix B for specific requirements.

The IITRAN/360 language and teleprocessing system is particularly well adapted to a secondary school situation. From its inception it has developed within an environment of secondary school computer science education. The original design included adequate control, by the local school, of both intentional and unintentional excessive use of computer time. It is a complete system. It includes a computer language specifically designed for student use, a teacher training program, a self-instruction textbook written primarily by secondary school teachers, and a technical specifications manual.

In the event that it is not feasible to install a telephone/computer interface, IITRAN/360 can be added to the library of programs on an IBM 360 and used in ordinary batch processing, along with other languages such as PL/I, Fortran, Cobol, etc., on-site and without teleprocessing.*

A MODEST BEGINNING

Operation COMPU/TEL is an outgrowth of a program which began five years ago at IIT. In the spring of 1962, Dr. Peter Lykos, professor of chemistry and director of the IIT Computation Center, addressed the twelve-member computer club of a suburban high school. Instead of giving the students the usual forty minutes of general information, he offered instead to repeat for the club the same experience he had given routinely to IIT juniors in physical chemistry. This involved a three-hour introduction to computers. The students would then write a computer program to process their own chemical data.

High School Proposal

Dr. Lykos realized that high school students were fully able and highly motivated to learn about and to use a modern computer. That fall he proposed to the IIT administration that IIT offer a similar opportunity to a larger number of high school students. His proposal for accommodating 150 students was accepted, and letters of invitation were sent to 200 high schools in the Chicago area. Almost 700 applications were received. The class size was then doubled in order to accommodate 300, and in Saturday classes students learned to program an IBM 1401 computer in machine language. Two of the high school students actually improved an IBM multiply subroutine. The high school program was designed to teach the students what a computer is and how one can use it--nothing more--as there seemed little possibility for follow-up experience in their own high schools.

But the high school teachers had yet to be reckoned with. They, too, wanted a computer education program. In the fall of 1963, an additional program was started for teachers. That year 170 teachers were trained in the IIT workshop. Those successfully completing the course were invited to select up to ten of their own high school students to be taken through an orientation meeting at IIT, to learn to program IIT's computer via a self-instruction programmed text, and then to use IIT's computer to support a computer-based program in the high school, under the teacher's supervision.

* Equipment requirements to use IITRAN/360 with and without teleprocessing are listed in Appendix B.

After five years, approximately 1000 teachers have participated in the IIT secondary school computer program at some level and have learned how to program IBM computers.

CURRENT SATURDAY PROGRAM

Courses for Students

The student program, as it has evolved, consists of an eight-course sequence, which is organized on three levels.

The first level contains one course -- High School Student Algebraic Compiler -- which meets from 9 a.m. to 3 p.m. on four Saturdays, and has an enrollment of 150 to 200 students. The primary objectives of this course are: to introduce the students to the world of computers, acquainting them with the capabilities and limitations of the device; to give students an introduction to IITRAN/360, an IIT-developed, low-cost, student-oriented computer language; to prepare them for writing programs; and to qualify students for subsequent courses. Basic skills and techniques of programming are taught, and mathematical background needs are kept to a minimum. This course is offered five times each year.

The second level also contains one course -- High School Student Assembly Language -- which meets on five Saturdays from 9 a.m. to 3 p.m., and has an enrollment of 50 to 100 students. In this course the student is introduced to a machine-oriented language, bringing him into closer contact with the logical structure of the digital computer. The concept of the stored program is stressed. Each student has the opportunity to process several problems, some of which are assigned, and some of his own choosing. This course is offered four times each year.

The third level contains the remaining six courses. Four are language courses and two are applications courses. All courses at this level meet for ten Saturdays, either from 9 to 12 a.m. or from 12 to 3 p.m. Enrollment in each is from fifteen to twenty-five students. At this level, provision is made for the differing interests of the students. Each student may select his own sequence of the following offerings:

1. FORTRAN -- A computer language designed for scientific applications

2. COBOL -- A computer language designed for business applications

3. SNOBOL -- A computer language adaptable to non-numerical problems, such as game playing, language analysis and translation, and the study of artifical intelligence

4. PL/I -- Programming Language I, the latest language developed by IBM for the 360 computer

5. Computer Applications - Simulation -- Lectures on model building, critical path method, matrices, and other selected topics

6. Computer Applications - Matrix -- Linear programming methods and matrix techniques

These courses are offered twice each year. A student who completed all of these courses would be quite knowledgeable in the computer field. In the past five years, approximately 10,000 students have taken the basic course, and many have gone on to advanced courses.

Courses for Teachers

The teacher program consists of six course offerings. The basic course is a six-Saturday computer workshop which meets from 9 a.m. to 3 p.m. Lectures in the IITRAN/360 language are followed by laboratory sessions to provide teachers with the opportunity to write and process their own programs. The IITRAN/360 teacher workshop is offered three times each year.

Four additional courses are available to teachers who wish to continue to broaden their experience with computers. FORTRAN, COBOL, PL/I and introduction to assembly language are offered twice each year, and meet for five Saturdays from 9 a.m. to 12 p.m. or from 12 p.m. to 3 p.m. Two courses may be taken simultaneously. Teacher programs are processed on the IBM 360 computer.

The most recent expansion of the Saturday computer program for teachers recognizes the need to provide industrial arts teachers with a computer experience related to the future of their field. Accordingly, a course designed and conducted in cooperation with Control Data Corporation in the area of symbolic control is now available. Symbolic control involves control of such devices as mechanical plotters and machine tools. This is accomplished through APT (Automatically Programmed Tools). APT is a computer programming system which prepares instructions for numerically controlled machine tools. Using the APT language, one may specify the operations involved to produce a part. The APT system translates these instructions into numerical information, which is interpreted by a control system to guide the machine tool through the motions required to produce the part. Teachers attend this course for ten Saturdays from 9 a.m. to 12 p.m. A first-hand experience of the complete process is provided. The APT course is offered twice a year.

OTHER DEVELOPMENTS

To go beyond an "introduction to computer programming" requires course and curriculum development. This is in rather a primitive state, but cooperative ventures are being initiated between the high schools and IIT.

One of the instructors in IIT's Saturday courses has shown a considerable interest in teaching IITRAN/360 to elementary school students. He has found that the idea is quite feasible. For more than two years he has regularly introduced the basic ideas of IITRAN/360 to sixth graders in two hours or less. These students, for the most part of average ability, are quickly able to construct simple programs involving the basic arithmetic operations.

Success

Since 1962 the IIT Secondary School Computer Science Program has enjoyed a large and increasing success. This is evidenced by the fact that to date some 1000 teachers and approximately 10,000 students have given up several Saturdays, have paid modest fees, and have received no academic credit for participating in this program. Enrollments continue to increase. This is probably due to three principal factors:

1. From the very beginning, the high school teacher was drawn into and made a part of the evolving program;

2. The program evolved in stages, from a simple, modest beginning, with continuous self-examination and revision. The teachers' comments and criticisms (and hence their students' comments and criticisms) were elicited and studied through personal contact, through group evaluation seminars, and through a newsletter edited by, contributed to, and distributed among the participating teachers; and

3. The IIT administration cooperated in making available needed resources, and offered the leadership of interested members of its faculty to the secondary school community.

The following excerpts are from report 67-754 to the Chicago Board of Education:

"COMPUTER SCIENCE EDUCATION
TO THE BOARD OF EDUCATION OF THE CITY OF CHICAGO:
THE GENERAL SUPERINTENDENT OF SCHOOLS

REPORTS that the pilot program at Lane and South Shore High Schools in Computer Science Education, as approved by the Board of Education, on June 22, 1966 (Board Report 74037-G), has met with great success and is now ready to be implemented in other secondary schools. The additional schools are: Austin, Bogan, Chicago Vocational, Crane, Dunbar, Lake View, Morgan Park, Schurz and Senn."

"THE GENERAL SUPERINTENDENT OF SCHOOLS

RECOMMENDS that in September, 1967, classes in Computer Science be continued at Lane Technical High School and at South Shore High School. In addition to these schools, Computer Science should be introduced to Austin, Bogan, Chicago Vocational, Crane, Dunbar, Lake View, Morgan Park, Schurz, and Senn High Schools."

Time for Action

In summary, IIT's experience shows that the cost has fallen low enough and the ease of programming has advanced sufficiently to make secondary school computer education practical. Interest and motivation on the part of secondary school administration, faculty, and students is sufficiently high to favor the success of such programs. We therefore believe that it is timely and desirable to embark upon comprehensive cooperative programs of course and curriculum development relating to the impact of the computer on secondary school education. The question is no longer <u>can</u> the computer be used in elementary and secondary school, but rather <u>how</u>.

APPENDIX A

THE IITRAN/360 LANGUAGE

Description

IITRAN/360 (Illinois Institute of Technology Translator) is an algorithmic language designed for student computer programming. The lang age syntax and the compiler's diagnostic capabilities facilitate rapid learning and easy programming and debugging. The IITRAN/360 compiler is capable of processing large numbers of student programs in a very short time, thus providing economical access to a computer.

The IITRAN/360 language was developed at IIT by William S. Worley.

Summary of Features

I. Easy and quick to learn

   A. Clear and natural language syntax

      1. Lack of arbitrary restrictions

         a. Modes may be mixed
         b. Multiple statements on a line
         c. Identifier length to 80 characters
         d. All 80 columns used for statements
         e. Format statements not required
         f. Nested IF ELSE statement
         g. Nested and n-dimensional subscripting

      2. Optional format specifications

      3. Simplified input/output for arrays

      4. Simplified operations on arrays

      5. Cross-section features

      6. Data types

         a. Real
         b. Integer
         c. Complex
         d. Logical
         e. Character

      7. Single and double precision

8. Program preparation

    a. 026 keypunch
    b. 029 keypunch
    c. teletypewriter

B. Clear and concise diagnostics

    1. Compilation diagnostics
    2. Execution diagnostics
    3. Trace
    4. Possible implementation of diagnostic ON statement for user error recovery

C. Semi-forgiving compiler

D. Execution always proceeds until occurrence of one of the following:

    1. Attempted execution of statement found erroneous by compiler
    2. Execution error
    3. Program completion

II. Cost Reductions for many classes of problems

    A. Machine cost -- fast compilation
    B. Programming cost -- minimal debugging time -- ease of programming
    C. Learning and programmer training costs

III. Applications

    A. Student programming
    B. Algorithm testing

Documentation

IITRAN/360 Self-Instructional Manual and Text, Addison-Wesley Publishing Company, Reading, Massachusetts.

IITRAN/360 Technical Specifications, published by IIT.

(An earlier version of IITRAN, also fully documented, exists for the IBM 7040.)

APPENDIX B

EQUIPMENT REQUIREMENTS FOR IITRAN/360

I. Card read input and printer output only:

IBM 360, model 30, 40, 50, 65, or 75 with 128 K (or more)
Memory protect
Universal instruction set
Disc operating system
    IBM 2540* Reader/Punch
    IBM 1403* Printer

II. Additional equipment required for IITRAN/360 teleprocessing system:

A. At Computer Center:

IBM 2702+ interface with telegraph type 2 terminal control unit
IBM 2311 disk drive (minimum of two with disc packs)
Data sets (103A)
Line adapters
Additional discs for foreground-background

B. At Local School:

One 33ASR teletypewriter with data set
Telephone line

---

* or other IBM standard input/output devices which DOS supports.

+ or 2701 or 2703.

APPENDIX C

A SUGGESTED ONE SEMESTER COURSE IN COMPUTER MATHEMATICS

AND PROGRAMMING FOR HIGH SCHOOL STUDENTS

I. OBJECTIVES

A. To introduce digital computer concepts to secondary students
B. To present the history, applications and social implications of computer technology
C. To indicate how problems can be solved on a digital computer and to examine the capabilities and limitations of the computer
D. To introduce students to a computer language so they may write and prepare programs for solving problems with a computer
E. To encourage students to develop facility with computers for future use in employment or study in various disciplines
F. To examine career opportunities that exist in the computer science field
G. To provide actual programming experience and validation of computer programs through access to a 360/40 IBM computer by way of remote telephone consoles

II. MAJOR AREAS OF INSTRUCTION

A. Introduction to Computers

1. Digital and analog computers
2. A brief history of computers
3. Anatomy and functions of a computer
4. The role of a compiler
5. Software
6. Employment and education
7. Present uses of computers
8. A look to the future of computers

B. The Flowchart

1. Steps in problem solving
2. Branching
3. Looping
4. Modifications of instructions

C. Introduction to the IITRAN/360 Language

1. The IITRAN/360 character set
2. The Assignment statement
3. Exponentiation
4. Order of Operations
5. Parentheses
6. Assignment of statements in sequence

D. Control Statements

1. Output
2. Input
3. Labels
4. Branching
5. Looping

E. Programming Experience

1. Analyzing the problem
2. Constructing the logic
3. Coding
4. Preparing input
5. Execution and debugging

F. Solutions of Mathematical Problems

1. Algorithms
2. Iterative processes
3. Polynomial evaluation
4. The slope function and applications
5. Polynomial approximations of circular functions

G. Statistical Applications

1. Measures of central tendency

    a. Mean
    b. Median
    c. Mode

2. Measures of dispersion

    a. Mean deviation
    b. Standard deviation

APPENDIX D

SAMPLE PROGRAM

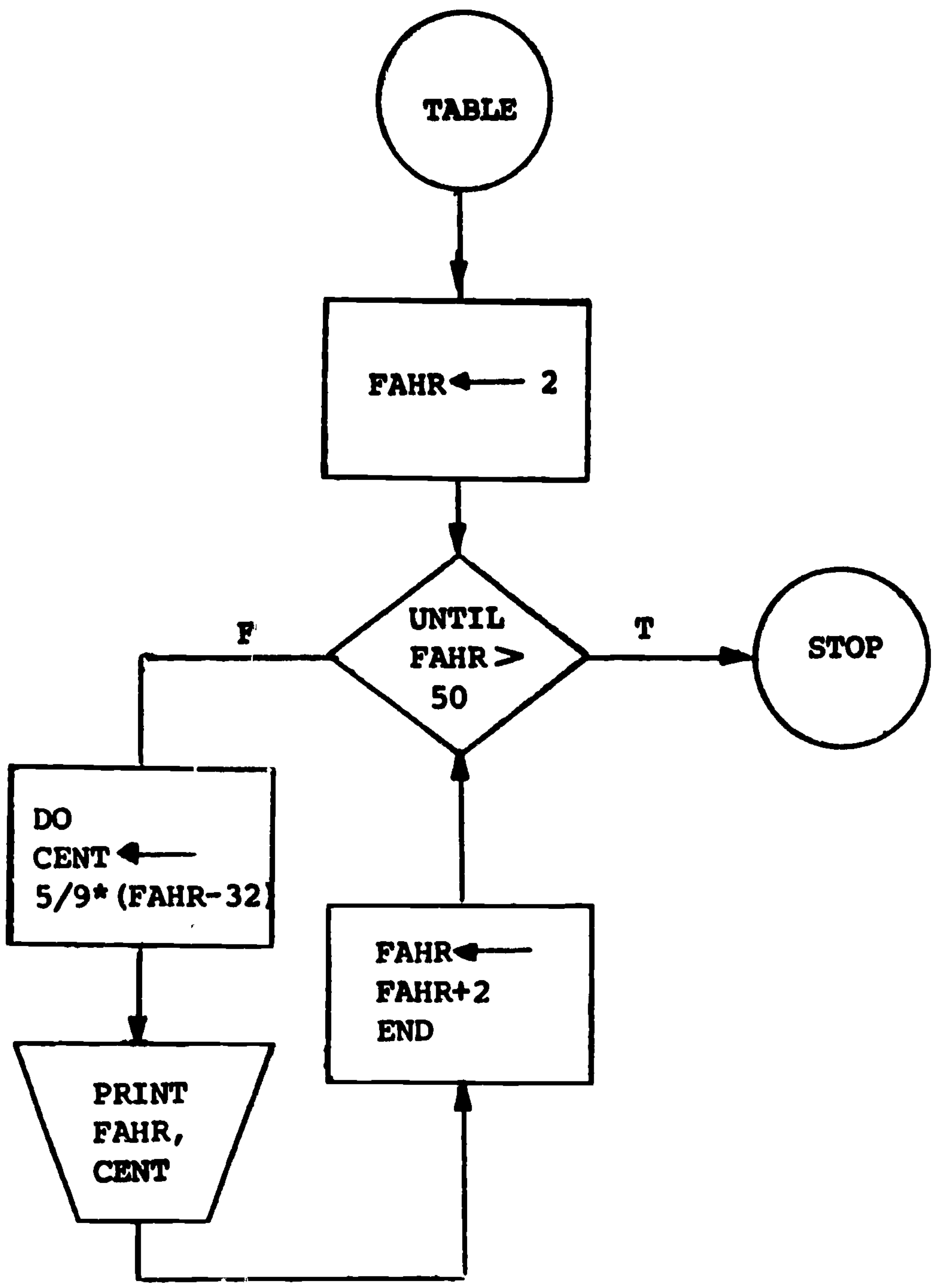
A sample computer program written in IITRAN/360 and processed via Operation COMPU/TEL:

Fahrenheit to Centigrade table
The formula for converting Fahrenheit values to Centigrade is:

$$C = \frac{5}{9}(F-32)$$

Write an IITRAN/360 program that will generate a two column table showing, in the first column, all the even Fahrenheit values from 2 to 50 inclusive, and in the second column, each corresponding Centigrade value.

Student Constructs Flowchart:



Types Instructions on Teletype:

```
// EXEC IITRAN
PROCEDURE TABLE
PRINT, 'BARRY ROSEN'
PRINT[4,2],'FAHRENHEIT CENTIGRADE'
INTEGER FAHR
FAHR ← 2
UNTIL FAHR >50
DO CENT ← 5/9*(FAHR-32)
PRINT, FAHR[10],CENT[12,2,1]
FAHR ← FAHR+2 END
STOP
END TABLE
EXECUTE TABLE
/*
```

Receives Output:

```
BARRY ROSEN

     FAHRENHEIT            CENTIGRADE
          2                 -16.66
          4                 -15.55
          6                 -14.44
          8                 -13.33
         10                 -12.22
         12                 -11.11
         14                  -9.99
         16                  -8.88
         18                  -7.77
         20                  -6.66
         22                  -5.55
         24                  -4.44
         26                  -3.33
         28                  -2.22
         30                  -1.11
         32                  0.
         34                  1.11
         36                  2.22
         38                  3.33
         40                  4.44
         42                  5.55
         44                  6.66
         46                  7.77
         48                  8.88
         50                  9.99
                              STOP STATEMENT OR END OF PROCEDURE ENCOUNTERED
                              AT STATEMENT 0011
```

EDUCATIONAL COMPUTER UTILITIES

BY

Herbert Bright

INTRODUCTION OF HERBERT BRIGHT

By Herbert Ohlman:

Our next speaker is Mr. Herbert Bright, president of Computation Planning, Inc., a firm engaged in systems analysis, design, management studies, research on computer applications, and programming. Mr. Bright holds three different B.S. degrees--and I'll leave it to him to explain how he managed to do that--from the University of Michigan at Ann Arbor (I guess he had three competing advisors), and he holds an M.S. in electrical engineering from the University of California at Berkeley.

Before he went into business for himself, he was supervising scientist of the Computation Planning Section at Westinghouse Electric Corporation, and he has been director of the Programming Research and Development Department of Philco's Computer Division. Mr. Bright has been active in many computer user groups, particularly in the IBM user's group, SHARE, where he served as secretary, as vice president, and as a member of the executive board; and in the Philco user's group, TUG, as president. He has also been very active in the Association for Computing Machinery, as regional representative, secretary, and vice president; he is presently council member-at-large and chairman of the National Program Committee of ACM.

I have asked Mr. Bright to talk to us about the "Potential of the Computer Utility Concept for Education." Mr. Bright...

EDUCATIONAL COMPUTER UTILITIES

Mr. Bright:

Thank you, Herb. That introduction, I think, starts me off on another direction from that on which I thought I would embark with you this morning. May I just quibble a little bit on the immediate objective? In the first place, I would like to point out that you see here quite a different orientation from the earlier speakers, who I think are very distinguished and prolific workers in the use of computer technology. In a sense, you might think of Ellis, Schure, and Lykos as the "artist-authors", whereas my orientation is that of "designer".

My intention here today was to give you first a brief survey of the use of computing in the educational community as it looks to a designer and producer of some of the tools that these people use, and also to give you just a smattering of the hardware and software mechanics associated with the new-looking computers, the so-called third-generation machines which permit all things to be happening concurrently.

My purpose in giving you this tutorial is not to make you into a mechanic--like myself and my associates--but rather, as much as anything, to remove some of the aura of mystery that has been associated with the subject. I'd like to at least define some of the widely used terms, so that the next time a computer salesman speaks to you about multiprogramming or multiprocessing, you can stop him in his tracks, and make him describe what's special about <u>his</u> multi-whatever.

Ten years ago, there was just beginning a significant use of the computer as a support tool in our institutions of higher learning. I recall having seen at a very early period that it was extremely difficult for people to get even what we would now think of as quite primitive computing tools. It was difficult not only financially, but operationally; neither the dollar support nor the application technology for educational use was available.

The situation has changed dramatically on both scores, as I'm sure everyone here is aware. This is especially true at the large-university level.

A trend that is becoming more and more visible is the use of computing facilities in smaller colleges and secondary schools in the educational process itself as well as for conventional administrative chores.

The three earlier speakers today have talked about the use of computers in the educational process in various kinds of relationships.

May I see a show of hands of those attendees here who have had at least some introduction to computer concepts, perhaps at the level of a series of tutorial lectures?

Well, I would judge that about two-thirds of you indicated that you are not completely innocent of any background in computer concepts. That is highly encouraging. I hope that I can subtract today from the remaining sum total of such innocence.

Since most of you are familiar with the basic concepts of digital computing, let's skip the traditional banalities about the processor, memory, and input/output devices. I ask you first to look at Fig. 1, which introduces a somewhat more generalized version of that usual basic system. Here we have generalized the number of all of the elements of the system except one: there is a single large memory, all of which can be accessed by any of the processors. The adjective "anonymous" applied to the processors means that any of the processors can be applied to any computing task or any part of a computing task. From now on I'll use the word "job" to mean any single assignment to a computing processor.

One characteristic of the so-called third-generation machines is that most of them, at least in principal, have been generalized so that they can be put together like the system of Fig. 1. Whether you want to do so is a question we'll discuss--with considerable shallowness--later.

Fig. 2 introduces the concept of the memory map, a bar-chart representation of jobs in memory. In Fig. 2 we show a system that can process two jobs at a time (although each processor can only deal with one job at a time) because there are two processors. At the instant of the snapshot map shown in this slide, each processor is busy with its own job and neither of them is being used by the Executive Program.

We'll refer repeatedly to the "Executive Program". This is a control program the function of which is to perform two basic tasks: it <u>dispatches</u> jobs, controlling the sequence in which each is handled; and it <u>schedules</u> the use of system resources such as memory space, processors, and I/O devices.

In Fig. 3, which let's look at next, we see a single-processor system that has somewhat greater generality of control than a basic system or than the one we just looked at. In this case, several jobs can be in active status in memory at one time. Although there's only a single processor, so that true simultaneous operation of more than one job cannot occur, we can have more than one job in partly-completed form and ready for further processing. This means that if, say, Job 1 is delayed because it needs access to equipment that is busy at the moment the need becomes evident, the processor need not wait or "hang", because it can immediately start working on another job that does not need that particular equipment. In particular, if a job is delayed because it needs access to an input/output device or to auxiliary memory, any other device that needs only the service of the processor can immediately start (or restart) being computed upon.

Both the dispatch and the scheduling functions of the Executive Program are required in order to decide which job can be serviced next and, of those that are eligible, which should be serviced next. Thus, the Executive gets into the act every time a job is stopped or delayed or a new job is to enter the system.

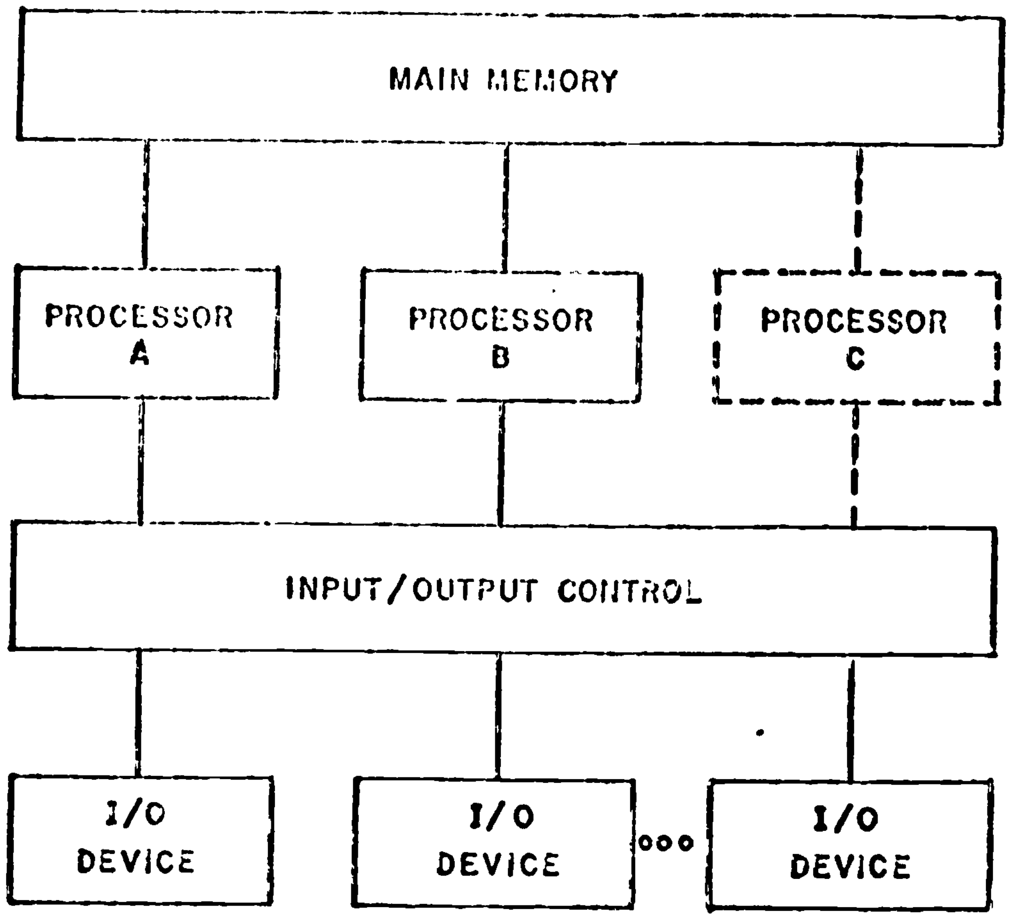


Fig. 1: Anonymous Processor Pool Concept

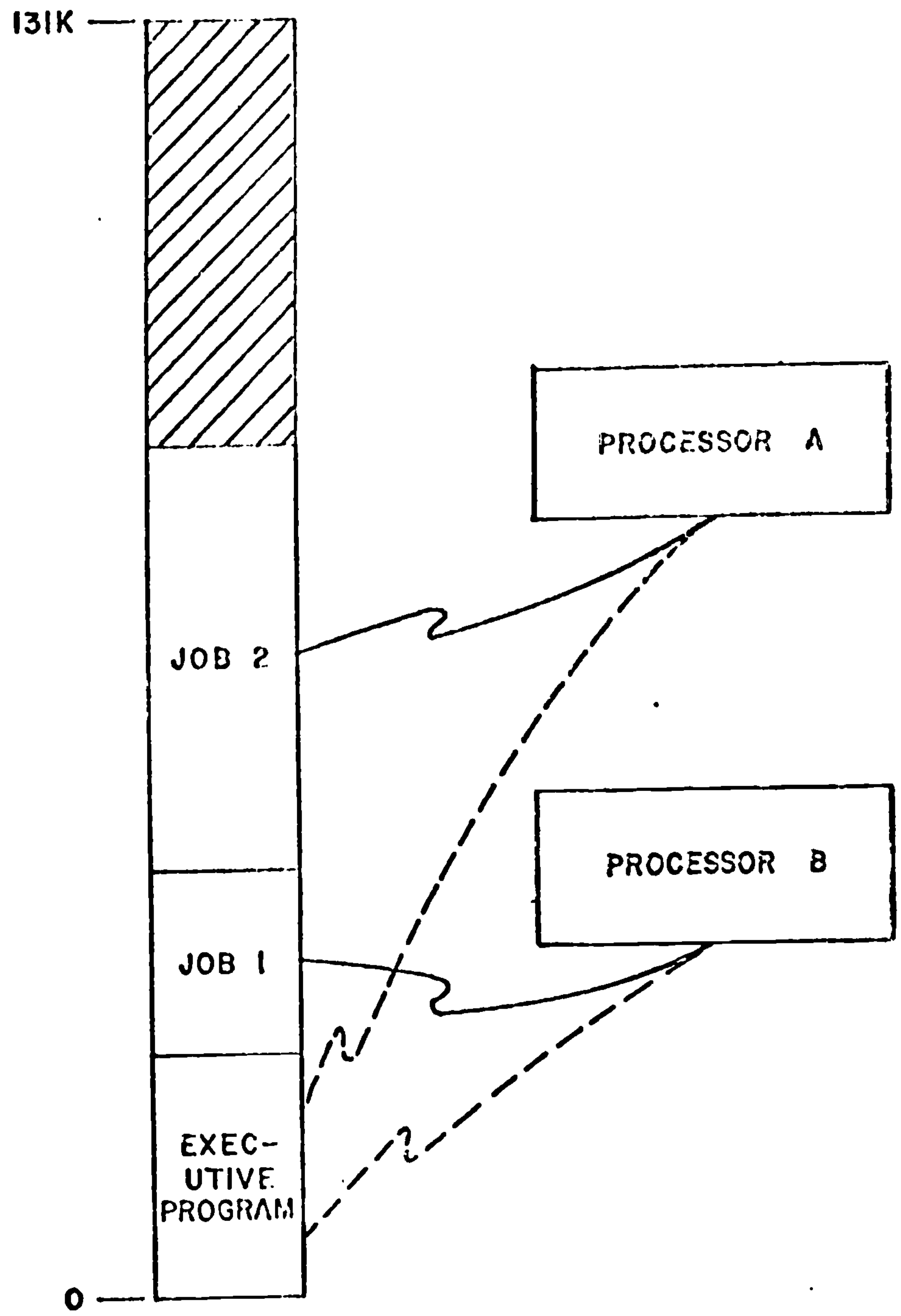


Fig. 2: Multiprocessing System (without Multiprogramming)

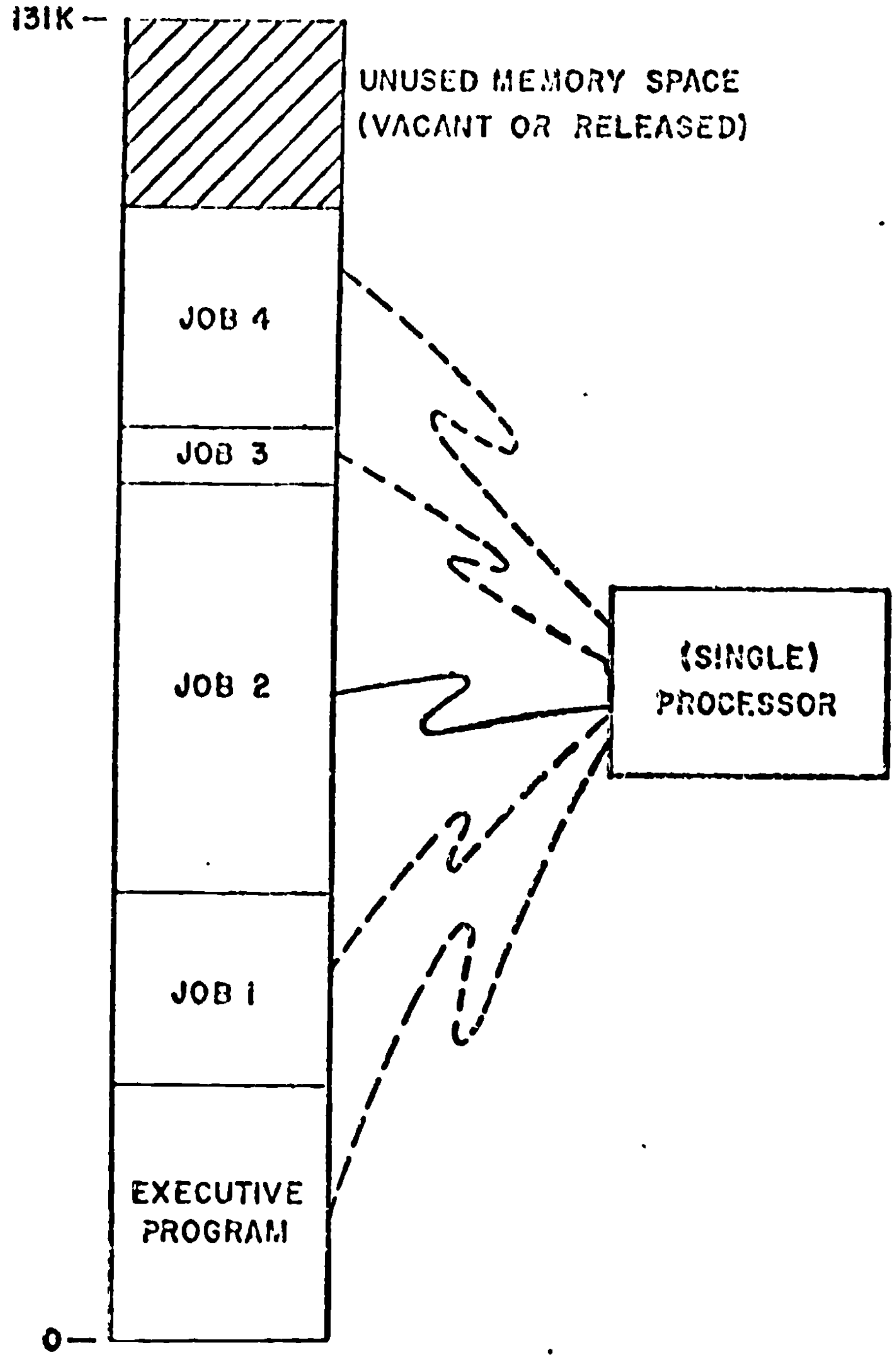


Fig. 3: Multiprogramming System

At the instant of Fig. 3, the processor is working on Job 2.

In Fig. 4 we see a somewhat more complicated situation that combines the two previous ones. We have a multiprogramming system, and there are two processors. At the instant of Fig. 4, Processor A is working on Job 13 and Processor B is performing an Executive function.

A typical sequence of actions for the processor to take when it enters the Executive Program would be something like the following: first, scan the table of jobs now in memory to see which are awaiting service; of those which need devices; of those which devices are available now, that is, are not already busy; of those that could be connected now, which should be connected next, taking into account how long the job has been waiting, how urgent it is, and how large or long its use of this device will be; having performed the foregoing dispatch function on as many jobs as can be brought from quiet into operating status, scan the table of jobs in memory to see if any have completed; update the memory assignment table to show that any completed job memory spaces are now available for assignment to new jobs; now scan the table of jobs awaiting entry to the system to find which can fit into available memory space; load one or more eligible jobs; now scan the table of jobs in memory, as done before, to determine if any job previously passed over should now get service from a device; if so, initiate such service; now scan table of jobs in memory again to see which should get service from a processor; now assign itself to that job and exit from the Executive Program.

If you got the idea from that oversimplified scenario of a typical sequence of actions within the Executive Program, that the executive's task is complicated, you are right. It is sufficiently complicated, in fact, that a really comprehensive multiprogramming executive system may consume a surprisingly large part of the available processor time. In extreme cases, this may be as much as half of the time. It also takes a lot of memory space.

You may well ask: "Is multiprogramming worthwhile if it is so costly of the two prime assets of the computer system--processor time and memory space?"

That's a good question. A part of the decision-making process when judging system proposals is the evaluation of costs and values.

Some multiprogramming systems consume less than ten percent of processor time and save more processor time than they consume by avoiding processor "hang". In some cases, the requirement to keep several jobs ready for instant computing may be urgent enough to justify poor economy.

When I started this talk, I said I'd remove the mystery from several buzz words that are popular in computer lore. Let's now edge up to a couple of more complicated concepts, paging and page turning. In so doing, we should get a look at some aspects of the important concept of memory protect.

We now have on the magic screen Fig. 5, which is a set of five successive snapshots or maps of memory contents after various Executive Program actions have been completed. You will note that these are just the memory map part of

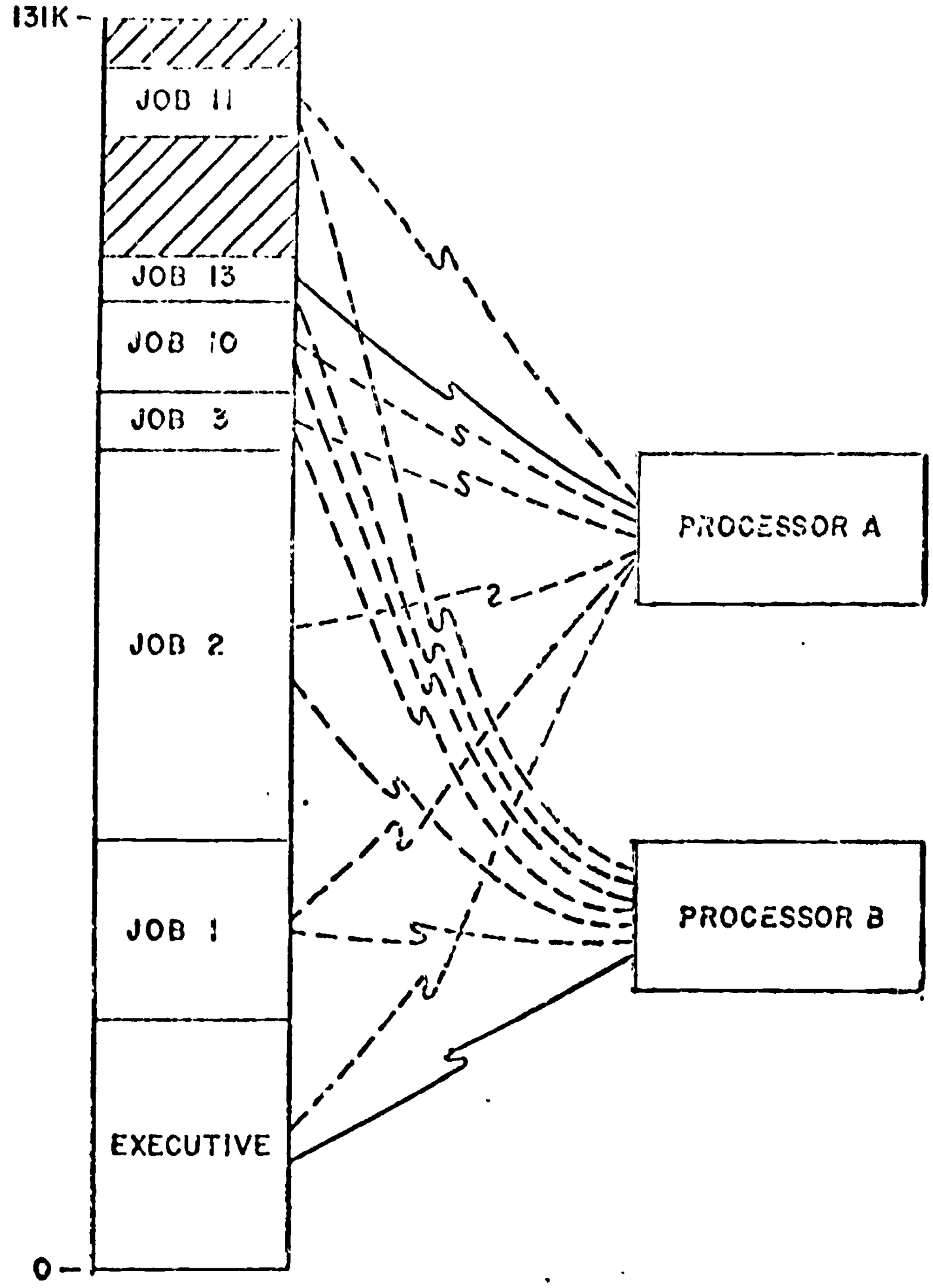


Fig. 4: Multiprogramming-Multiprocessing System

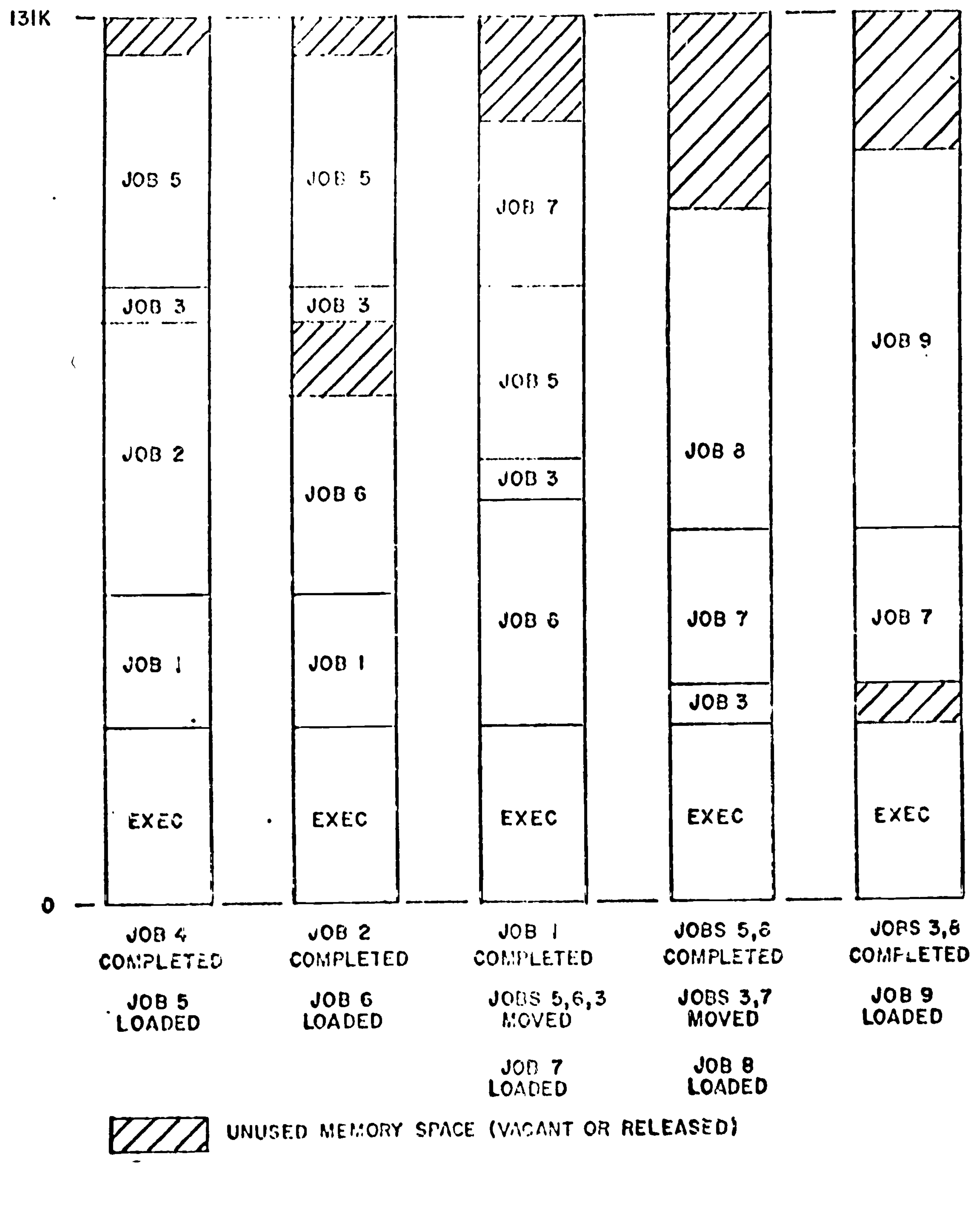


Fig. 5: Multiprogramming System Memory Maps

the previous slides, with the processor or processors not shown. This is appropriate, because we are now going to talk about the management of memory space as such, without regard for processor assignment, in a multiprogramming system.

In the first map, Jobs 1, 2, and 3 remain in memory from previous actions. Job 4 has just been completed, so its space was no longer needed; consequently, Job 5 was loaded into the space previously occupied by Job 4.

In the situation displayed by the second map, Job 2 has been completed; Job 6 has been loaded into that space.

Note that Job 6 took less space than Job 2 had taken, so that there is some vacant space between Job 6 and 3. This situation would proliferate if we didn't do something about it, so we would have lots of such vacant spaces. In the next snapshot the system does do something about it.

Now look at the third map. Job 1 has been completed and its space was available for use. It appears that no waiting job could be loaded into any of the sections of memory that were then available; however, the sum total of empty space was large enough to contain Job 7; consequently, the system moved Jobs 6, 3, and 5 in order to use up the small empty spaces and thus make available a single space big enough to accommodate Job 7, which was then loaded.

Let's stop for a minute and think about what just happened. The contents of maybe one-third of the memory was moved elsewhere into memory in order to coalesce available memory space. This is a fairly time-consuming task, because in addition to reading and writing every item in most of memory, it is necessary to update a large number of Executive Program data tables to account for new locations of entire jobs and also to account for new values for some, but not all, of the symbols that translate into physical addresses in memory for each program that was moved. This kind of thrashing around is known as housekeeping, and can be quite time-consuming.

Furthermore, I need hardly do more than hint of the chaos that results if any single one of the actions involved in moving jobs to different locations happens to go wrong. This might happen as a result of a hardware error; it is much more likely to happen as a result of a software error.

Let's take one more step backward. You might think that software, such as the Executive Program itself, should be free of error, that is, perfect, before it is released for customer use.

As a member of the software-building fraternity, let me assure you that except in the case of trivial items of software, this is almost never true and only approaches truth for major software items, such as executive systems when they have been in service for several years by many users. (Remark: I have said, often enough to be tiresome, when told that a particular software item is now "clean," that "You have found its last bug if you are now ready to scrap the program!")

Why, you ask, can't a major software item really be "clean"? The answer is that the number of possible paths through the logical maze that constitutes the decision-making part of a major executive program is so huge that all possible combinations will probably never be tested.

Sooner or later a condition will be encountered for which all of the alternatives have not properly been provided. When that happens, disaster strikes. For the benefit of those of you who are not computer buffs, let me tell a simple story that will illustrate such a disaster.

In an executive system produced under my supervision while I was with a computer manufacturer, one subroutine caused a magnetic tape unit used for system input to read one block of data, which was always of fixed length, into what is called a "buffer area," which is a section of memory used to temporarily hold data.

This particular subroutine had a fault in it, in that the program neglected to test for completion of the action of reading the block of data into memory. This test should have been performed before the system assumed that the reading action was complete, as is the case with many seemingly minor actions in a computing system.

Well, the program ran successfully for three years without this fault having been detected. It so happened that the system had other tasks that were always performed before the data that had been brought in were used, and these tasks permitted the input action to be completed by the time its result was needed.

Then we produced a much faster system that used the same magnetic tape units and was, except for speed, almost perfectly compatible with the older computer processor. When we attempted to use the executive program on the new system, it was unable to successfully start its first program execution. The reason was quickly evident.

The faster computer completed those "other tasks" before the tape unit had completed reading the first block of data. Consequently, the system started to use data from the buffer area that was partly new and correct, and partly old information from previous usage of the buffer area.

Thus, in spite of the fact that that part of the executive program had been used successfully several hundred million times, the presence of a serious error in the software was not discovered until a major change of system timing occurred.

This particular horror story is really not a serious case, especially since its result was nice and obvious. In a major piece of system software, there unfortunately may be lots of errors that are not at all obvious until lots of thoughtful people have made lots and lots of use of the system with that software in service.

Let's take another look at Fig. 5. The last two snapshots show more of the same, with jobs completing, jobs being moved as required, and so on.

If the moving of jobs is so costly and tricky, how about finding ways to avoid the need for moving jobs in memory? Well, needless to say, some thought has been given to this problem and it has affected the development of both hardware and software.

The procedure I described above was an all-software scheme, that is, the executive system did whatever was necessary in order to move the jobs and change appropriately all subsequent references to those jobs.

One way to speed up the operation of such memory moving and at the same time simplify the software would be to provide hardware to translate the addresses of all references to a given job at the time it is used.

In principle, this requires automatic translation by hardware that would otherwise be done by accessing data in memory. The obvious way to do this is through use of indirect addressing, which is a feature of many modern computers (but not the IBM 360 series).

The obvious objection is that this takes considerable time: for each address that is indirect (that is, that contains the location of an address rather than the address itself) an additional memory cycle or two must be used every time memory is accessed.

There are other problems. Chief among these is that certain control actions should be associated with the memory address translation process. These include not only memory-protect (that is, denial of access where access might cause trouble or is at least restricted), but the taking of different actions in different situations.

For these reasons, which also complicate the all-software approach, memory-address-translating hardware is fairly complicated in function and high in price.

One approach to controlling excess cost is by reducing some of the generality. As is often done, the scheme shown in Fig. 6, which I now show you, does this by translating only part of the length of the address. Such a scheme is known as "paging," which will follow from an examination of how it works.

Think of each address, which is a number denoting the physical location of each memory cell, as a number consisting of two parts: a page number, as in a book, and a word number within each page.

Taken together as a single number, the page and word number parts of a single address identify a single cell in memory.

Suppose now that we change the number of a page. In effect, this relocates a given memory cell to a different apparent location in memory.

The system shown in Fig. 6, which is one I proposed in 1964, performs this "paging" action by translating only eleven bits of a 24-bit address. The left-hand eleven bits of an address from a program are used to identify any one of 2048 locations in a special high-speed memory containing page numbers and also containing certain information relating to the control of access to each page.

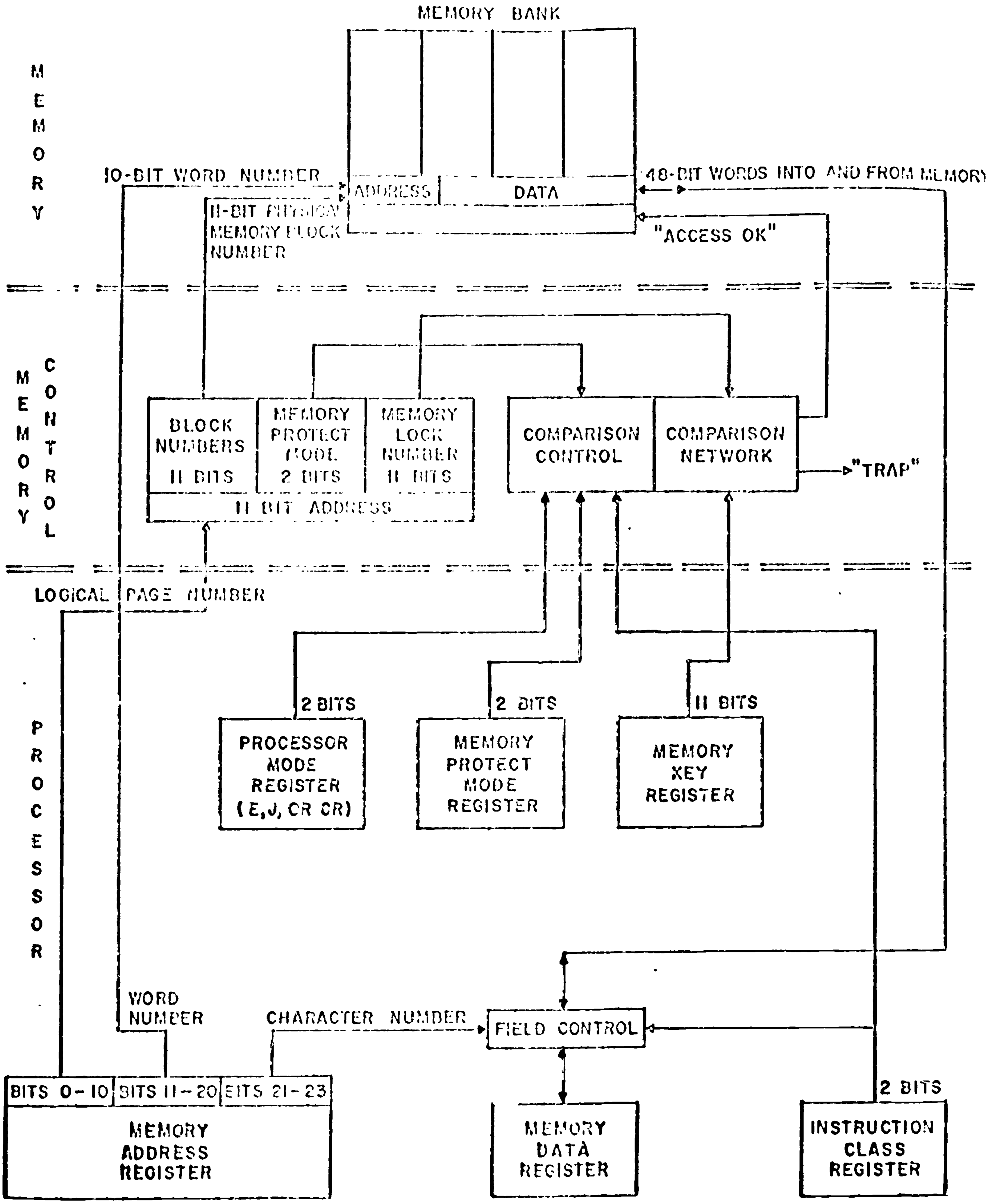


Fig. 6: Memory Paging and Protect Features

Using only part of the address in the translation process does two things: it reduces the amount of data to be translated for each word accessed, and it reduces the number of address items that need to be stored. This second item is the important one: in the system shown, the 24-bit address can identify any of eight million items; however, the 11-bit page number field needs only a couple of thousand locations to hold all possible translation numbers. In practice, this partial-address translation causes some efficiency loss but is not prohibitively inefficient; full-length address translation would be prohibitively costly.

I won't drag you through the details of the memory-protect scheme. I will say only that features are provided to perform two kinds of protection against unauthorized or accidental use of particular pages. If you're interested, you can find an explanation, in tiresome detail, in a paper I presented to the 1964 Fall Joint Computer Conference.*

So much for a description of what is meant by the jargon item "paging." Now, as I promised you about ten minutes ago, I'll say a few words about the companion but quite different buzz-word, "page-turning." I will do this without pictures, but hopefully in less than a thousand words.

Several years ago, when our English cousins (who usually scoop us by a year or two and seem to do everything with more zip and less money than we, with our army-ant approach to technological problems, seem to need) started to look at the problems of running many parts of large programs concurrently in machines that for each could only assign small memory allocation, they invented the concept of thinking of pieces of programs as "pages," the idea being that only one page of program and one page of data should be in the computer's fast memory at a time. This concept applies only to true multiprogramming systems.

Whenever the program transferred control to a part of itself that was not in the "page" presently in memory (we now call this "makes an out-of-page jump"), or required access to data outside the data page now in memory (we now call that "makes an out-of-page fetch"), the processor was to stop work on the program it was working on and initiate action to bring the required page into memory. The system would perform the input action needed, possibly performing a related output action in order to make space for the needed page in memory, meanwhile starting work on some other task.

Bringing the required page into memory, together with the related housekeeping actions, came to be called "page-turning." This concept is an important one and is the subject of much debate and development effort. Unfortunately, automatic page-turning, if we consider practical system and program parameters, often implies unacceptably low efficiency of system use. Consequently, this concept is of more importance in the so-called computer-

---

* "A Multiprogramming/Multiprocessing System," in 1964 Proceedings, American Federation of Information Processing Societies, part II, pp. 97-141, Spartan Books, 1965.

science community than in the much larger world of computer users at the present time. New hardware developments may change this but it's true right now.

I'd like now to turn to different kind of topic, but one that is gaining a lot of attention in the educational community at the university level. This is the idea of graceful system degradation, or fail-soft operation.

This concept originated with the military and aerospace organizations, who had some computer programs that simply could not be allowed to collapse even if a part of the computing system failed. They were willing to have the computer system reduced in capacity or effective speed and, of course, were willing to have the system take a short time to recover its equilibrium after a hardware fault. What they couldn't stand was a complete crash, or a so-called "drop-dead failure."

The concept is taken from the basic philosophy of management of military organizations that has been with us at least since the days of the ancient Greek phalanx: slow down or retreat, but don't collapse.

I was surprised a couple of years ago to discover that the people at the University of Michigan were quite serious about being willing to spend substantial money to avoid having a computing center, that may be serving hundreds of users at a given time, collapse in the event of failure of a unit of hardware. Since then, I have found this interest to be shown by many other advanced computer users in academe. You should at least know what it means, because you, too, may be faced with the problems to which this concept offers a partial solution.

The basic idea is that the system must be completely redundant as to hardware, and must be organized so that it can take automatic defensive action. Let's look at Fig. 7.

Ignoring minor details, you'll note that there are at least two of every kind of system unit. Furthermore, and importantly, there is even redundancy of system wiring and switching. That is, each memory switch is wired to, and can handle, all of memory, and the same holds for each input/output switch. Not visible here is the fact that certain vital control tables of the Executive Program are duplicated in different memory units.

Suppose, now, that there is a major system failure, such as loss of correct action by a unit of memory, that is detected by routine tests of the hardware that are performed at frequent intervals under control of the Executive Program.

The following actions occur under control of the Executive Program: First, the Executive Program tables are checked for apparent correctness, by performing appropriate software tests on the tables in the surviving part of memory. Tables that appear to be questionable in correctness are flagged so they will be used only with appropriate caution.

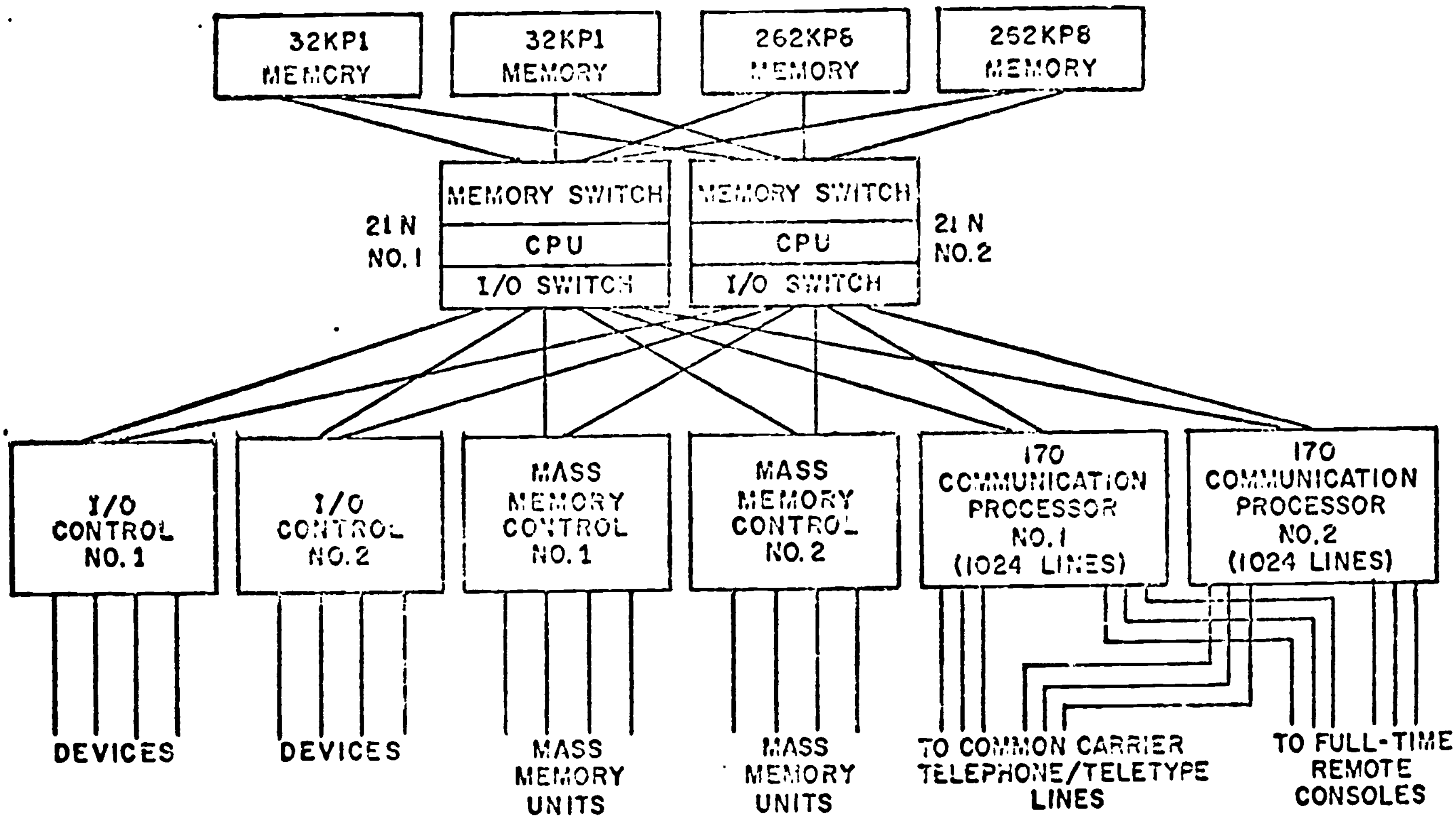


Fig. 7: Redundant System

Next, the tables of system resources, including memory space, are updated to show that all of the space within the defective memory unit is no longer available.

Next, all jobs currently in the system are examined for priority and special management considerations, such as those relating to personnel safety. This means that more of the tables of data in the Executive Program's controlled areas are examined. Mandatory jobs are flagged, and as many other jobs as can still be handled with the reduced present system capability are appropriately identified.

The system then resumes action at the new (lower) level of capability, having taken into account the resources available and the tasks to be handled.

So much for the fancy-sounding concept of graceful degradation: all it costs is money!

To finish out my stint in speaking to you from the viewpoint of the computer mechanic, I'd like to introduce one final idea to you: this is the concept of common routines, also known as re-entrant routines, pure procedures, and single-copy routines.

The basic idea is that, in a many-user system, there will often be many of the users who are in the system at any time who may need to use the same general-utility programs.

Thanks to the fact that, about fifteen years ago, the computer world ignored the preachings of Professor Howard Aiken of Harvard University, who had insisted that programs and data should be stored separately in different memory hardware, we now have a problem: programs and data being stored together, programs can be modified by themselves or by other programs. When programs are modified purposely, this is considered by some to be clever and can in fact be useful; when they are modified by accident, the results are always bad.

In order to avoid this kind of trauma, in conventional programming there is a basic restriction that only one user or one "calling program" can use a given program or subprogram at a time.

In the case of large rograms, such as compilers that may be in demand by many users at a time in the same system, we thus have a serious problem: there must be a separate copy of each such program for each user who needs access to it. I recall that several years ago, Professor Allan Perlis, now of Carnegie-Mellon University, told me he had discovered that on one occasion the main memory of the large Carnegie computer--one of the earliest true multiprogramming systems--happened to contain seven copies of the ALGOL compiler. This experience convinced him that what he chooses to call "single-copy programming" is necessary for large educational systems.

Fig. 8 illustrates the nature of control flow between two calling programs (identified as "Program A" and "Program B"), which may move around in memory, and three subroutines that may be used by both programs concurrently. Notice that Subroutines 1, 2, and 3 are present only one time each and do not move between usages.

I will not drag you through the details of what housekeeping operations are necessary in order to preserve correct operation in the face of this degree of generality in the way subroutines are called. I will merely identify some of the kinds of restrictions that must be applied, and some of the system requirements.

Common routines must not be modified during use. All "scratch areas" (that is, memory used for temporary data storage during a calculation) must be related to the location of the start of each calling program. These requirements together dictate that common routines must contain no data that may be modified in use; all such data must be stored elsewhere.

The system must be capable of automatically correcting the control linkages between calling programs and common routines. It is desirable (but fortunately, not mandatory) that subroutines nested to arbitrary depth be capable of being accessed, linked, or exited in arbitrary sequence.

The most widely-used present term for this kind of capability is "re-entrant programming." It is becoming important principally in that several manufacturers claim to have re-entrant compilers (that is, compilers that can be used simultaneously by many users) in test use at the present time. In one educational system being developed by my firm, we have chosen to avoid use of this technology at least until it can be demonstrated to be mature enough and efficient enough to be used for large-scale computing.

You have all been very patient indeed while I maundered through some of the ideas that lie behind the present "multi-everything" computing systems. I should like now to talk a bit about the ways such systems are actually accessed by remote users.

There are two basic access concepts that describe all present remote-access educational computing systems: time-sharing or interactive computing, and batch-processing.

Batch processing is the older approach. In it, computing tasks are presented, and results are received as single or monolithic input and output data files. Its name stems from the way in which the earliest executive systems were loaded with programs in groups or "batches." Originally, because of hardware and software limitations, such systems were slow in response (that is, had "long turnaround time"), partly because of the fact that individual programs could only be processed in the order in which they were presented to the system. Modern batch system technology permits flexible sequence control (that is, short and urgent programs can be processed before long and non-urgent programs) and, consequently, permits quick response (short turnaround time) for small jobs.

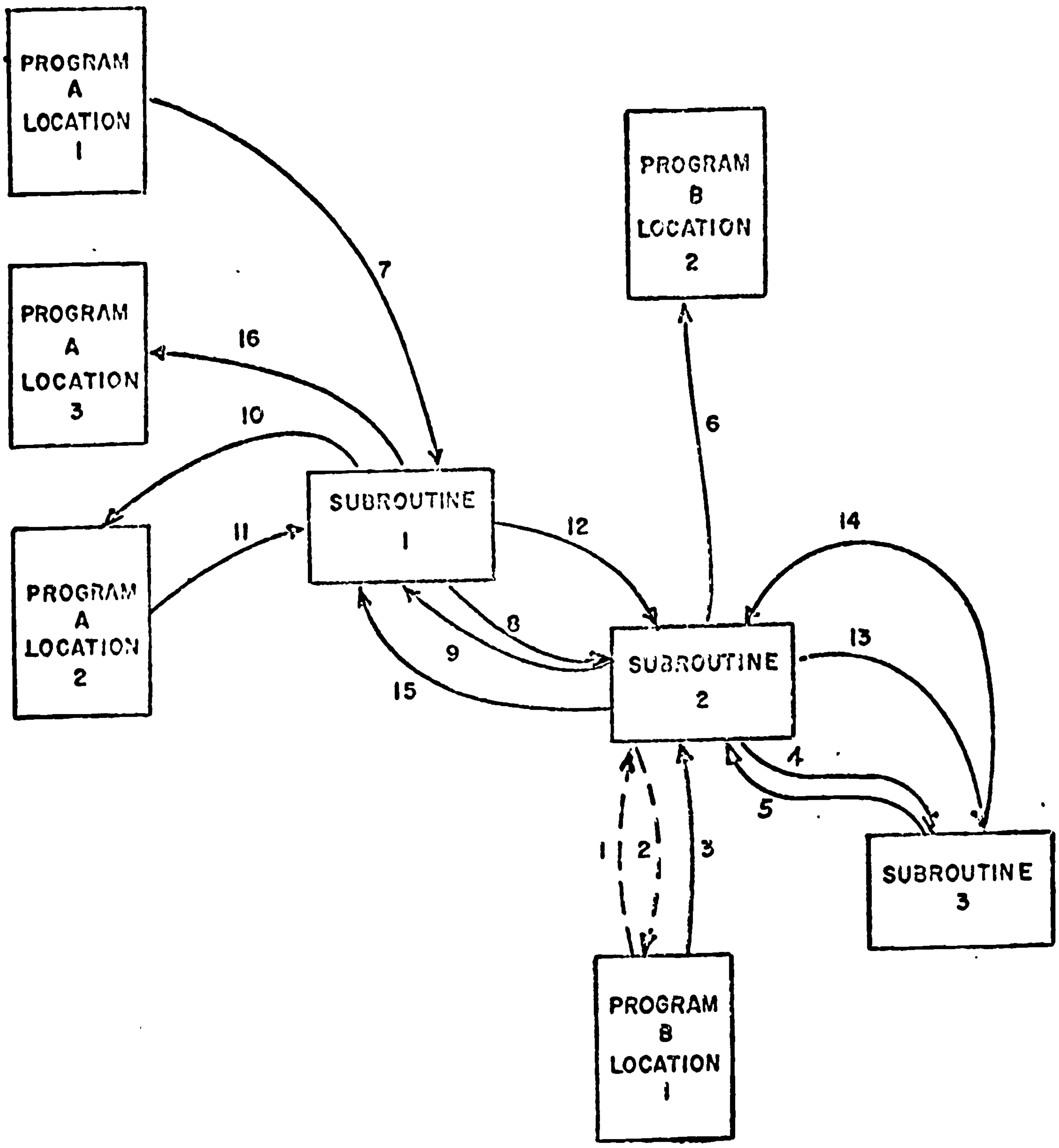


Fig. 8: Common Subroutine Linkages

Time-sharing systems permit users to communicate with the system continuously during the preparation and execution of programs. They provide, in a sense, a removal of "turnaround time" as a system variable; because a user is "on the system" all of the time he is active and the results of program changes may be very quickly evident. Such systems have disadvantages for all users, as well as the obvious advantage of good communication between every user and his developing program. The disadvantages for economically-limited systems, such as most educational systems, include limited access to input/output terminals (which must be available to a user during all of the time he is working on his program), low-speed input and output and substantially higher system overhead cost (that is, more of the system's resources are used to support the access process itself).

For advanced users and for high-priced users such as engineers and scientists (whose salaries are much higher than the equivalent cost of computer terminal usage), the values of the time-sharing approach may be very high and the costs may be acceptable; in some cases, the apparently high cost of such service may represent an exceedingly good "buy" in that the service can substantially increase the productivity of entire organizations.

For elementary or beginning users, the values of the conversational interaction between user and system have not been shown to be large. For student users in economically limited circumstances--which include those in many institutions of higher learning and most students in secondary schools--the costs of time-sharing service are difficult to determine but seem to be substantially higher than the batch-type service.

There is a special advantage in batch-type service, unrelated to cost: the user does not have to do his creative thinking and perform clerical work on his program while he is using an expensive terminal device (and by so doing, he is preventing others from using it).

I feel that modern multiprogramming system technology permits good turnaround time for large numbers of small-scale users of major systems such as those used to support the teaching of large numbers of students. I also feel that such technology permits the lowest possible cost for student service.

I know that many educational systems now in use or being planned use the time-sharing approach. I do not know that significant evidence exists to support a conclusion that time-sharing is substantially more efficient in the pedagogical sense, that is, that students learn faster or better when provided with interactive computing as compared to monolithic computing service ... at least to the extent of the same dollar-cost limits.

All of the discussion I have given you for the past half-hour has related to the announced subject of this talk, that is, the so-called computer utility in education. I have talked about some elements of the technology and about the two basic methods of access relating to the use of large systems by large numbers of small-scale users.

There is another idea that has gained well-deserved attention in the educational community: this is the idea that a small-scale, very-low-cost computer may be used to serve a small number of users. If the large-scale computer system is analagous to an electric utility, the mini-computer concept is analagous to having your own rope-started electric generator in the back yard.

Hardware and software developments during the past three years in the manufacture and distribution of mini-computer systems have shown that the cost per small user can be very attractive ... perhaps even competitive with the use of the large (or, as we have called them, "computing utility") systems. Furthermore, many complications and uncertainties associated with large systems that are "not yet fully operational" have properly caused concern on the part of educational administrators.

I feel that the advantage of the utility approach is that of generality. The large systems provide access to large quantities of high-powered software and to the work of other highly-skilled users, permit the use of techniques that require large memory or huge numbers of computational steps, and permit coordinated activities among users and comprehensive and consistent administration of system usage.

An example of the applicability of the large system is that of wide usage of important programs, such as those constituting a typical computer-managed-instruction system, of the kind described by Dr. Schure. In at least one potentially important use in school administration, the use of the large system is required (for effective application of the techniques of school resource scheduling), although most administration tasks could be performed well on small machines.

As I fade slowly into the twilight, I'd like to say that I think the utility systems, using old technology brought up-to-date, will be important in education, and that your understanding of computing technology will be important to you as well as to education. Thank you.

DISCUSSION: HERBERT BRIGHT

Question: Do you essentially need a double machine in order to have a fail-soft situation? You emphasized this yourself--but if cost is going to be an important factor, you have to weigh the desirability of fail-soft against the six breakdowns a day that you have in a single system for a given power.

Reply: That is correct. I'd say, like most other decisions that I'm aware of in planning and acquisition and use of computer systems, there are a large group of cost-value tradeoffs. In defense of this position, I'd like to point out that as the number of users increases, the probability is substantial that there will be at least one, or perhaps several, who could be badly hurt by irrevocable loss of part of the results. A loss of just one completed result derived from a file by a doctoral candidate (supported in some of his research by the military) occurred recently. This loss was caused by an operator error. In effect, one unduplicated item overwritten by some other information caused the loss of the equivalent of several hundred thousand dollars of Defense Department funds, and probably required another six to nine months to reconstruct the information. And this happened to only one graduate student.

There are situations in which there are a few hundred or a few thousand users in operation concurrently, such as service to, for example, a city hospital. You don't have to think entirely of losses occurring, let's say, if a large rocket got out of control at launch (in that case, there are more than numbers involved), but I think as the scope and the depth and the quantity of applications in the student department increases, the probability of unacceptability of complete loss of power to the system goes up. We have got to depend on this thing.

Herbert Ohlman: This is another sign that computer technology is becoming more like a public utility. If you remember the big power failure in New York, and what that did to a great city, you can see by analogy that if we do get all of our records and much of our control information into a system of this sort, failure could be catastrophic.

Question: Can you discuss some applications in the area of computer graphics?

Reply: Most people here are generally familiar with the concept of the "numbers racket": large-scale arithmetic processing and problems represented by numbers having significance; and also with the processing of large volumes of non-numeric information, such as index and abstract files in a library support organization. What is not so obvious is the processing of other kinds of information, such as where the meaning of information in the file is concerned with spatial relationships. The field of graphics, which has become very active in the last few years, consists of the processing by a hardware-software complex of information describing spatial relationships between elements. The best reference I know of is Ivan G. Sutherland's 1959 doctoral thesis at M.I.T. on "Sketchpad"--the first large-scale application of a machine to this kind of graphics problem.

A great deal of attention has been given over the past several years to the ability to describe information related to the spatial relationships of items. An example which has gotten a lot of publicity is the creation of computer-animated cartoons. Then, there is a scheme for machine-aided design in which the designer works on a two-dimensional display with a device which allows him to relate what appears on the display with the information content of the machine. With use of such a device (a so-called "light pen" is frequently used), the computer can sense the presence or absence of light spots on the screen. Also, there is usually a keyboard to insert alphabetic and numeric information, and possibly knobs or two-dimensional ball-type controls, by which a user can change the inter-relationship of objects.

For example, he can "sketch" a three-dimensional object on the screen and then instruct the system to consider the roughly drawn lines to be straight ones, and the system will perform this operation. He can also specify the dimensions of each line. Thus, the system can take a graphic description of this object, with its dimensions, and the user can get back a conceptual view of how the object looks from different directions. By instructing the system to perform three-axis rotations, he can see different spatial orientations, with the system performing all the bookkeeping necessary to adjust the several stages of projection of this conventional descriptive geometry. Furthermore, the user can insert a little object into a larger one; he can design a certain element and then cause it to be replicated many times into a more complex structure.

The point of all this is that much of the brute labor that has in the past been involved in the design of complex geometric shapes and other kinds of entities by engineers and other designers can now be supported by a large processing system. It is not hard to think of the effectiveness of an individual person being multiplied by thousands of times, because once having created the information, he can process it by demand. In the limit, one can think of the results of the session in a machine-aided design project turning out to be instructions to numerically-controlled milling machines, cam cutters, and so on, to actually create a new machine from a few key presses and strokes, so that we may have a system which can dramatically extend the operating capability of an individual. There is incidentally, in the November 1967 issue of American Education, a photograph showing some of the things I have described.

Peter Lykos: I mentioned before that most of our concern has been with non-vocational areas, but as long as graphics was brought up, we did institute as part of our Saturday program for teachers something which is aimed at those vocational teachers concerned with mechanical drawing and with machine tools. Last fall we instituted a Saturday course for teachers dealing with the APT (Automatically Programmed Tools) language as an example of symbolic control. What they have to face up to is that people that they are now training in conventional drawing and tooling are probably going to be obsolescent by the time they graduate and certainly will be obsolete in the not too distant future.

On the point of three-dimensional representation, I saw a presentation made in a Bell Laboratories symposium in which they brought out that a geometry teacher normally works with a two-dimensional chalkboard. He draws a square (and we all accept what a square is); but then he talks about a cube--but it's difficult to represent a cube, so we use a projection of a cube on a two-dimensional surface. What the people at Bell Labs did was take advantage of stereoscopic viewing and moving at the same time and talked about a hyper-cube--that is, a cube which is a four-dimensional object projected into a simulated three-dimensional space.

Herbert Ohlman: Incidentally, in this area of graphics, from the standpoint of possible changes in the curriculum, I remember a few years ago while I was with Xerox Corporation, that their interest was in computers, communications, and graphics--particularly graphic communications. We decided that one of the best areas to expand into was engineering graphics. I went over to the local university to find out what they were teaching in this field, and the head of the department said that they were not teaching it at all any more, beyond a very basic course.

Herbert Bright: Incidentally, I'd like to talk about a few surprises I got last week in visiting an installation which has, I think, one of the most effective multiple-user support systems in operation. It operates in a batch-mode, in which the user program is taken and processed and tossed back to the user. In a large organization with large problems, queues build up (such as Pete described), resulting in long turnaround times (the amount of time lapse between the request for service and the receipt of results in the hands of user). The turnaround time may be anywhere from hours to perhaps days. Needless to say, it's quite a thing to go from a seemingly instantaneous intellectual process to a wait of hours or days--hence, the instant appeal of the immediate-response devices with which one can literally communicate with his problem. However, there is another side of the story, and that is where we have many typical students who turn out thousands of simple programs. One would like a system for handling such users to have a turnaround time of a few minutes.

Well, last Thursday I happened to visit the Westinghouse Research Laboratories where they have a very highly polished system for small user programs. A priority scheme penalizes those users who are too greedy in machine time or memory space or input/output. Most of the users arrive at the machine with a problem very much like what we feel is typical for secondary school or introductory university level student programs. I tried it myself. I walked into the middle of a large crowd by the counter, behind which there were a couple of technicians serving a two-processor system with a multiprogrammed multiprocessor. I had my results 3 minutes and 31 seconds after I dropped the request on the table. This wasn't unusual as far as this organization is concerned, and I went away shaking my head because it was so much better than the factor of 50 that I was used to in comparable organizations. I think the point here is that if a system is in fact planned to favor the small user, then the old slow-batch concept can, in fact, be made to come back pretty fast. Perhaps it may even turn out to have some kinds of advantages over time-sharing, where the user may feel under continual pressure--with 12 people waiting behind you for a console--that he may be inclined to be sloppy because he must push things to get off, rather than stay and think. Here we have a concept that may turn out to be very significant, because it clearly is going to be less costly to process a user task that is presented in one lump to the processor than it is to process the task in 200 pieces, with the systems support required for such a number of interruptions. So I must mention to you that while it has become almost a cliche in this business that for good service we need so-called multiprocessors, again I think we're beginning to see that perhaps more careful, more effective use would go a long way to solving the problem.

Peter Lykos: Next time you're in Chicago, stop in at the Museum of Science and Industry. There is a teletypewriter in the Illinois Bell Telephone Exhibit. These people have come up with a demonstration--sort of at random, so we never know when they're going to come at us--and they have a script, and partway into the script, it requires them to dial up the IIT Computation Center and send in a trivial problem (which is, I think, a length conversion--from inches to centimeters). About two minutes later, they dial up and ask for the processed results to come back to them. We use a remote job entry system which does not involve multiprogramming or multiprocessing--a fairly modest 360 model 40. However, we do have an automatic scheduler built in, and what we've done--because we know that their problem is a trivial one, and because the demonstration is essential--we give them top priority, so that whatever program is running, their's will be the next one processed. And because we too process large numbers of small jobs, we support some 1200 undergraduates at IIT, in addition to all the secondary school people. The scheme that you're talking about is exactly what's implemented--high priority is given to the very short jobs.

CLOSING REMARKS

by

Herbert Ohlman

CLOSING REMARKS: HERBERT OHLMAN

I hope you'll agree with me that today's session has been well worthwhile. I'd like to thank the many people who have been involved in organizing this Symposium, particularly Andy McCormick, head of our Data Processing Center, and Don Thomsen, Assistant Director in charge of our Memphis office--who together with me dreamed up the idea for this symposium--and also the many people who have helped out in the planning and execution of it-- Glen McAlister, head of our Educational Materials Center; Dan Magidson, our Audio-Visual Specialists; Verna Smith and Jim Dean, who have handled publicity for us; and our staff of secretaries, particularly Charlotte Trudo, Stephanie Macks, and Leslie Osborn. Also, I'd like to thank our other Assistant Directors who suggested attendees and helped us in planning; Earl Morris in Illinois, Boyd Carter in Kentucky, and Don Thomsen and Jim Winter in Tennessee.

I want to thank you all for coming. Please feel free to talk to the speakers afterwards, or get in contact with us or the speakers after the meeting.

APPENDIX I: SYMPOSIUM ATTENDANCE

Attendees are grouped by state,
and then by municipality.

SYMPOSIUM ATTENDANCE

ILLINOIS

Dr. Leslie Woelflin
Southern Illinois University
Carbondale, Illinois 62901

Dr. Peter Lykos
Director of Information Processing Center
Illinois Institute of Technology
Chicago, Illinois 60616

Mr. A. R. Evans
State Dept. of Illinois
316 South 2nd Street
Springfield, Illinois 62701

Mr. Don C. Norwood
Office of Public Instruction
State Dept. of Illinois
316 South 2nd Street
Springfield, Illinois 62701

KENTUCKY

Dr. Boyd Carter
CEMREL, Inc.
408 East 15th Street
Bowling Green, Ky. 42101

Mr. O. A. Mattei
Supervisor
224 East Main
Bowling Green, Ky. 42101

Dr. Henry N. Hardin
Western Kentucky University
Bowling Green, Ky. 42101

Mr. Charles Bratton
State Dept. of Education
Frankfort, Ky. 40601

Mr. George Rush
State Dept. of Education
Frankfort, Ky. 40601

Mr. Jack Meisburg
Louisville Public Schools
Louisville, Ky. 40208

Mr. Samuel V. Noe
Superintendent
Louisville Public Schools
Louisville, Ky. 40208

Mr. Maurice Silvey
Louisville Public Schools
Louisville, Ky. 40208

Mr. Patrick McNeil
Hopkins County Board of Education
Madisonville, Ky. 42431

Mr. H. Carroll Hanberry
Federal Programs Coordinator
Owensboro Board of Education
Owensboro, Ky. 42301

Mr. Richard Brown
Paducah Independent School
Paducah, Kentucky 42001

Mr. William Whitlow
Paducah Independent Schools
Paducah, Ky. 42001

Dr. Newman Walker
Superintendent of Schools
Paducah Independent Schools
Paducah, Kentucky 42001

MISSOURI

Mr. Ronald H. Womack
Director of Data Processing
Parkway School District
455 North Woods Mill Rd.
Chesterfield, Mo. 63017

Dr. William Hedges
Clayton School District
7530 Maryland Avenue
Clayton, Mo. 63105

Mr. James Blumenburg
Radio Corp. of America
7710 Carondelet Avenue
Clayton, Mo. 63105

Mr. Glenn Einsel
Systems Representative
Radio Corp. of America
7710 Carondelet Avenue
Clayton, Mo. 63105

Mr. Carl VanDyke
Training Director
Radio Corp. of America
7710 Carondelet Avenue
Clayton, Mo. 63105

Mr. C. R. Hill
St. Louis Junior College Dist.
7508 Forsyth Boulevard
Clayton, Mo. 63105

Mr. John Alspaugh
University of Missouri
Columbia, Mo. 65201

Mr. Roger Harting
University of Missouri
Columbia, Mo. 65201

Mr. Charles W. McLane
University of Missouri
Columbia, Missouri 65201

Mr. Warren R. Seymour
University of Missouri
Columbia, Mo. 65201

Mr. Shannon W. Ennis
Ozarks Bi-Cultural Center
Eminence, Missouri

Mr. Erwin Neff
Ozarks Bi-Cultural Center
Eminence, Missouri

Mr. Donald R. Shire
Ozarks Bi-Cultural Center
Eminence, Missouri

Mr. Mel Reginald Sims
Ozarks Bi-Cultural Center
Eminence, Missouri

Mr. Harold A. Carmichael
Hannibal Public Schools
Hannibal, Missouri 63401

Mr. Larry Lutz
Hannibal Public Schools
Hannibal, Missouri 63401

Mr. Elmer Myer
Hannibal Public Schools
Hannibal, Missouri 63401

Mr. Harry F. Thompson
Director of Data Processing
Jefferson City Public Schools
609 Union
Jefferson City, Missouri 65101

Mr. Norris G. Pearson
Lincoln Lab High School
Jefferson City, Mo. 65101

Mr. W. Eugene Dabbs
State Dept. of Education
Jefferson City, Mo. 65101

Mr. Jack Roy
State Dept. of Education
Jefferson City, Mo. 65101

Miss Joan Bartel
Mid-Continent Regional
Educational Laboratory
104 East Independence Ave.
Kansas City, Mo. 64106

Dr. Eugenia Koos
104 East Independence Ave.
Kansas City, Mo. 64106

Mr. Warren Sarff
Assistant Superintendent
Kirkwood School District
516 South Kirkwood Road
Kirkwood, Mo. 63122

Mr. Homer E. Willis
Louisiana Public Schools
Louisiana, Mo. 63353

Mr. Gary D. Sharpe
Assistant Director
Title III Programs
St. Charles School System
1916 Elm Street
St. Charles, Mo. 63301

St. Louis

Sister Mary Aimee
Dilschneider
Academy of the Visitation
3020 North Ballas Road
63141

Sister M. Marcelline
Rubinelli
Academy of the Visitation
3020 North Ballas Road
63141

Mr. Don Burris
Addressograph-Multigraph
5240 Oakland Avenue
63110

Mr. Charles Rutledge
Addressograph-Multigraph
5240 Oakland Avenue
63110

Mr. Charles Pickett
Burroughs Corporation

Mr. Jim Dean
CEMREL, Inc.
63074

Mr. Robert Hanke
CEMREL, Inc.
63074

Mr. Beldin Hare
CEMREL, Inc.
63074

Mr. Dan Magidson
CEMREL, Inc.
63074

Mr. Glen McAlister
CEMREL, Inc.
63074

Mr. A. J. McCormick
CEMREL, Inc.
63074

Mr. Herbert Ohlman
CEMREL, Inc.
63074

Dr. Wade M. Robinson
CEMREL, Inc.
63074

Mrs. Verna Smith
CEMREL, Inc.
63074

Mr. Peter M. Hirsh
Florissant Valley Community
College

Mr. Gary L. McIrvin
Customer Applications Specialist
General Electric Company
1139 Olive Street - Room 500
63101

Mr. John Cheek
Hazelwood School District
63042

Mr. C. O. McDonald
Hazelwood School District
63042

Mr. Rich A. Kane
Honeywell
200 South Hanley
63105

Mr. Kent C. Mueller
Honeywell
200 South Hanley
63105

Mr. Thomas E. Yager
John Burroughs School
755 South Price Road
63124

Mr. Ron Garrett
Ladue School District
63124

Mr. Edward Greenwood
Ladue School District
63124

Mr. William Halbrook
Ladue School District
63124

Mr. Franklin P. Morley
Ladue School District
63124

Mrs. Nancy Heyde
McCluer High School

Mrs. Stella Farley
McCluer High School

Mrs. Mary Syptert
McCluer High School

Mrs. Marcella Wins
McCluer High School

Mr. Albert Cooper
Normandy School District
63121

Mr. Ed Potter
Normandy School District
63121

Mr. Mel Sheehan
Normandy School District
63121

Mr. Roland L. McCammack
Parkway School District
63017

Mr. Charles W. Avey, Jr.
The Principia
63141

Mr. Howard H. Buer
The Principia
63141

Mr. Clyde K. Cantley
Ritenour Consolidated School District

Mr. George F. Chapman
Ritenour Consolidated School District

Carmelita O'Connor
St. Louis Board of Education
63101

Dr. Milton O. Jones
St. Louis Junior College District
63105

Mr. Tom McClintock
Data Processing Coordinator
St. Louis Junior College District
7508 Forsyth Boulevard
63105

Mr. Anzo Manoni
Educational Consultant
St. Louis Post Dispatch
63101

Reverend Paul Kidner
St. Louis Priory School
500 South Mason Road
63141

Reverend James Daly, S.J.
St. Louis University High School
63110

Mr. W. D. Crawford
Scott-Foresman

Mr. Samuel Moses
Special School District
63119

Mr. Ronald Compton
University City High School
63130

Mr. Harris Jackoway
University City High School
63130

Mr. R. S. Halsey
University City School District
63130

MISSOURI (cont'd.)

St. Louis (cont'd.)

Mr. R. N. Yeager
University City School
District
63130

Dr. Robert Elsea
University of Missouri at
St. Louis
8001 Natural Bridge Road
63121

Mrs. B. R. Perkins
Visitation Academy
3020 North Ballas Road
63141

Warrensburg

Mr. Robert Grimes
Central Missouri State
College
64093

Mr. Richard Houerstock
Central Missouri State
College
64093

Mr. Jon Rickman
Central Missouri State
College
64093

Mr. John Willhardt
Central Missouri State
College
64093

Waynesville

Mrs. Ruth Anderson
Ft. Leonard Wood Schools

Mr. Robert Knight
Ft. Leonard Wood Schools

Mr. C. W. Parker
Ft. Leonard Wood Schools

TENNESSEE

Mr. Jack Carr
Chattanooga Public Schools
P. O. Box 2013
Chattanooga, Tenn. 37409

Mr. Ray Coleman
Chattanooga Public Schools
P. O. Box 2013
Chattanooga, Tenn. 37409

Mr. William B. Stradley
Tennessee Technological Univ.
Cookeville, Tenn. 38501

Mr. J. C. Sawyer
Principal
Dyersburg High School
Dyersburg, Tennessee 38024

Mr. Ralph Mays
Humboldt City Schools
202 North 14th Avenue
Humboldt, Tennessee 38343

Mr. William A. Sadler
Humboldt School District
2780 Jackson Highway
Humboldt, Tennessee 38343

Dr. Donald Thomsen
CEMREL, Inc.
3808 Norriswood
Memphis, Tennessee 38111

Mr. Ted McCloud
Memphis City Board of Edu-
cation
2597 Avery Avenue
Memphis, Tennessee 28112

Mr. E. Ray Holt
Memphis City Schools
2597 Avery Avenue
Memphis, Tennessee 38112

Dr. James Winter
CEMREL, Inc.
Nashville, Tennessee 37211

Mr. James H. Hogge
George Peabody College
Nashville, Tennessee 37203

Mr. Charles Pullen
Tennessee State Dept. of
Education
100 B. Cordell Hull Building
Nashville, Tennessee 37219

OTHER LOCATIONS

New York

Miss Ruth Callahan
Center for Innovation
New York State Department of Education
Albany, New York 12224

Dr. Erwin Gordon
270 Westend Avenue
New York, New York 10023

Dr. Alexander Schure
New York Institute of Technology
Old Westbury, New York 11568

Metropolitan Washington, D.C.

Dr. John Lehmann
Staff Associate
National Science Foundation
20550

Mr. James C. Gillis
President
Quality Educational Development, Inc.
1820 Jefferson Place, N.W.
20036

Mr. Douglas Rabel
Quality Educational Development, Inc.
1820 Jefferson Place, N.W.
20036

Mr. Al Coulson
Radio Corp. of America
20006

Mr. Richard Otte
Bureau of Research
U.S. Office of Education
20202

Mr. Frank Schmidtlein
Bureau of Research
U.S. Office of Education
20202

Mr. Herbert Bright
Computation Planning, Inc.
7979 Old Georgetown Road
Bethesda, Maryland 20014

Other

Mr. A. Buford Moulton
State Dept. of Education
Tallahassee, Florida 32304

Dr. Harry Bowman
Southeastern Educational Laboratory
3450 International Blvd.
Suite 221
Hapeville, Georgia 30054

Dr. Allan Ellis
New England Data Systems
Cambridge, Mass. 02138

Mr. John Seger
Northwest Regional Educational Laboratory
Portland, Oregon 97204

Mr. Bob Willits
Appalachia Educational Laboratory
P. O. Box 1348
Charleston, West Virginia 25325