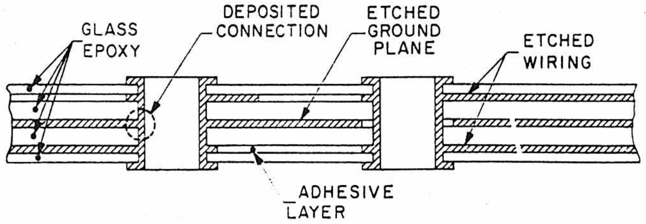

Fig. 12 Three Layer Etched-Wiring Cross-Section

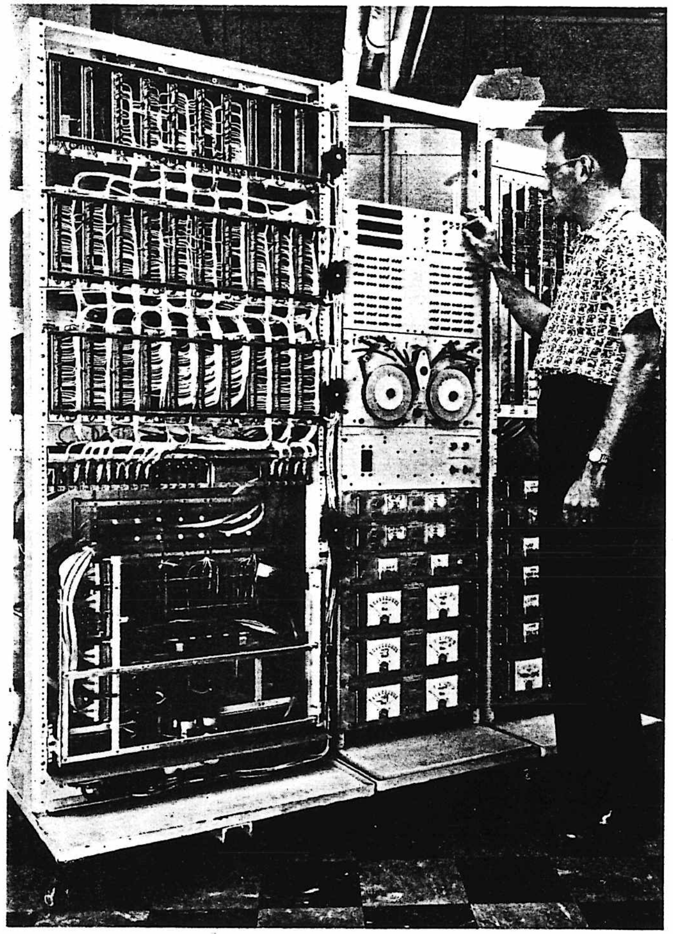

Fig. 13 Overall View of the FX-1 Computer

AFIPS 1962 Spring Joint

ON-LINE MAN-COMPUTER COMMUNICATION

J. C. R. Licklider and Welden E. Clark
Bolt Beranek and Newman, Inc.
Cambridge, Massachusetts and Los Angeles, California

Summary

On-line man-computer communication requires much development before men and computers can work together effectively in formulative thinking and intuitive problem solving. This paper examines some of the directions in which advances can be made and describes on-going programs that seek to improve man-machine interaction in teaching and learning, in planning and design, and in visualizing the internal processes of computers. The paper concludes with a brief discussion of basic problems involved in improving man-computer communication.

Introduction

On-line communication between men and computers has been greatly impeded, during the whole of the short active history of digital computing, by the economic factor. Large-scale computers have been so expensive that -- in business, industrial, and university applications -- there has been great pressure to take full advantage of their speed. Since men think slowly, that pressure has tended to preclude extensive on-line interaction between men and large-scale computers. Inexpensive computers, on the other hand, have been severely limited in input-output facilities. Consequently, the main channel of on-line man-computer interaction, in the world of commerce and in the universities, has been the electric typewriter.

In critical military systems such as SAGE, the economic factor has been less restrictive and the need for man-computer interaction greater or more evident. However, the SAGE System, the pioneer among computerized military systems, is "computer-centered" -- less so in operation than in initial design, but still clearly computer-centered -- and that fact has had a strong influence upon man-computer interaction in military contexts. The computers and their programs have tended to dominate and control the patterns of activity. The scope for human initiative has not been great. Men have been assigned tasks that proved difficult to automate more often than tasks at which they are particularly adept.

For the kind of on-line man-computer interaction required in computer-centered military systems, a console featuring a Charactron display tube, a "light gun," and arrays of display lights and push buttons proved effective. At one time, about four years ago, at least 13 different companies were manufacturing such consoles -- different in minor respects but all alike in basic concept. Until recently, therefore, on-line man-computer communication could be summed up in the phrase: electric typewriters and SAGE consoles.

Increasing Need for Man-Computer Symbiosis

During the last year or two, three trends that bear upon on-line man-computer interaction have become clear. First, the cost of computation is decreasing; it is no longer wholly uneconomic for a man to think in real time with a medium-scale computer. Second, time-sharing schemes are beginning to appear in hardware form; the economic obstacle fades as the cost of a computer is divided among several or many users. Third, more and more people are sensing the importance of the kinds of thinking and problem solving that a truly symbiotic man-computer partnership might accomplish:

1. Military officers are eager to regain the initiative and flexibility of command they feel they lost to the computers in computer-centered command and control systems, but they want to retain the storage and processing services of the computers.

2. A few mathematicians are finding computers very helpful in exploratory mathematical thinking. Working closely with powerful computers and graphic displays, they are able to see at once the consequences of experimental variations in basic assumptions and in the formulation of complex expressions.

3. Several persons responsible for the programming of computerized systems are beginning to believe that the only way to develop major programs

rapidly enough to meet hardware time scales is to substitute, for the large crews of programmers, coders, and clerks, small teams of men with sophisticated computer assistance -- small teams programming "at the console." With statement-by-statement compiling and testing and with computer-aided bookkeeping and program integration, a few very talented men may be able to handle in weeks programming tasks that ordinarily require many people and many months.

4. In war gaming and even to some extent in management gaming, there is a growing feeling that the value of exercises will increase greatly if the pace can be speeded. On-line interaction between the gamers and computers is required to speed the pace.

5. In the planning and design of systems of many kinds, digital simulation is recognized as a valuable technique, even though the preparation and execution of a simulation program may take weeks or months. There is now a growing interest in bringing the technique under direct and immediate control of planners and designers -- in achieving the availability and responsiveness of a desk calculator without losing the power and scope of the computer.

6. In the field of education, some of the far-reaching possibilities inherent in a meld of "programmed instruction" and digital computers have become evident to many.

7. The complex equipment used in exploratory research, now in scientific laboratories and perhaps shortly in space, requires overall guidance by scientists but, at the same time, detailed control by computers. Several groups are currently interested in "semi-automatic laboratories."

The foregoing considerations suggest that man-computer communication will be an active field during the next few years and that efforts to facilitate productive interaction between men and computers will receive wide appreciation.

Man-Computer Complementation

The fundamental aim in designing a man-computer symbiosis is to exploit the complementation that exists between human capabilities and present computer capabilities:

a. To select goals and criteria -- human;

b. To formulate questions and hypotheses -- human;

c. To select approaches -- human;

d. To detect relevance -- human;

e. To recognize patterns and objects -- human;

f. To handle unforseen and low-probability exigencies -- human;

g. To store large quantities of information -- human and computer; with high precision -- computer;

h. To retrieve information rapidly -- human and computer; with high precision -- computer;

i. To calculate rapidly and accurately -- computer;

j. To build up progressively a repertoire of procedures without suffering loss due to interference or lack of use -- computer.

It seems to us that the functions listed, a through j, are the essential ingredients of creative, intellectual work. In most such work, they are not strung together in simple temporal sequence, but intimately interrelated, often operating simultaneously with much reciprocal interaction. For that reason, the conventional computer-center mode of operation, patterned after that of the neighborhood dry cleaner ("in by ten, out by five"), is inadequate for creative man-computer thinking; a tight, on-line coupling between human brains and electronic computers is required. We must amalgamate the predominately human capabilities and the predominately computer capabilities to create an integrated system for goal-oriented, on-line-inventive information processing.

In associating capabilities a through f primarily with human beings and capabilities g through j primarily with computers, we are of course describing the present state of affairs, the technology in which we now must work, and not asserting any essential discontinuity between the domains of human and machine information processing. There is always the possibility that human competence in g through j can be significantly increased, and it is almost certain that machine competence in a through f will develop rapidly during the next decades. At present, however, we think that man and computer complement each other, and that the intellectual power of an effective man-computer symbiosis will far exceed that of either component alone.

Steps Toward Man-Computer Symbiosis

To bring men and computers together in tight synergic interaction, we must make advances in several contributory fields. Among the most important appear to be: time sharing and other possible solutions to the economic problem; memory and processor organization for contingent retrieval of information and programming of procedures; programming and control languages; and on-line input-out equipment, including integrated displays and controls. The groups with which we are associated have been working in those areas. It is disappointing to find that the areas appear to grow more rapidly than we can explore them and to realize how trivial are our accomplishments relative to the requirements. However, we are beginning to have some tangible results, and it may be worthwhile to illustrate briefly the following three:

1. A system for computer-aided teaching and computer-facilitated study.

2. A man-computer system for use in the planning and design phases of architectural and constructional problems.

3. Two programs that display aspects of the internal processes of a computer during execution of programs.

Computer-Aided Teaching and Learning

Exploration of ways in which a computer can facilitate teaching and learning raises several problems in man-computer communication. Effective teacher-student relations involve nearly continuous interchange of information, and anything that interferes with the communication is likely to impair effectiveness.

The importance of rapid, convenient student-teacher communication has demonstrated itself quite clearly in experiments with a simple, automated, language-vocabulary-teaching system. One version of the system, *Tutor 1*, uses a computer typewriter as the communication link between the student and the machine. Let us examine first the procedure briefly and then the problem of typewriter communication between student and computer.

The typescript of the sample German-English lesson, shown in Fig. 1, illustrates the procedure. In a session with *Tutor 1*, the student initiates activity by typing "O." The computer then asks him whether or not the student wants detailed instructions. The student replies by typing "s" for "No, start the lesson." The computer selects a German word at random and presents it. The student then types an English word that he thinks is equivalent in meaning and terminates his response by hitting the "centered-dot" key. If the response is acceptable to the computer, the computer types "+" for "correct." (Brevity is crucial.) If it wants another English equivalent, the computer then types the German word again. If it does not want another English equivalent, it types the item score and the cumulative score to date and offers a comment on the student's performance. When the student misses a word, the computer types "-" for "incorrect" and "ta" for "Do you want to try again?" If the student replies "y" for "yes," the computer presents the missed German word again. If the student replies "n" for "no," the computer types an English equivalent and requires the student to copy it. And so forth, as illustrated.

The first thing we found out about *Tutor 1* was that students (children and adults) who type well like to use it, whereas students who do not type well may be attracted at first but soon tire of the lesson. During the development of the program, several variations were tried out. Those that speeded the pace of presentation or streamlined the procedure of response were the most successful. A version that eliminated the requirement that the student type the response -- that allowed him to respond vocally or subvocally and then trusted him to score his answer -- was greatly preferred by students who typed only fairly well or poorly; good typists liked "type-the-answer" versions better. With one type-the-answer program, designed to avoid all possible interruptions, students who type well sat for two or three hours at a time, industriously adding new German, French, or Latin words to their vocabularies, occasionally checking their cumulative scores, but never asking for coffee breaks.

Twenty years from now, some form of keyboard operation will doubtless be taught in kindergarten, and forty years from now keyboards may be as universal as pencils, but at present good typists are few. Some other symbolic input channel than the typewriter is greatly needed.

We make some use of the light pen and "light buttons" associated with multiple-choice questions and answers displayed on the oscilloscope. When the alternative courses of action can be laid out in a tree-like branching structure, it is convenient to let the computer ask a multiple-choice question via the oscilloscope display and to arrange the program in such a way that touching the light pen to the button associated with particular response brings forth a subordinate question appropriate to that response. With four familiar alternatives, the operator can make a selection every second or two (i.e., select at a rate of 1 or 2 bits per second), which is adequate for some purposes, though not truly competitive with talking or expert typing (up to 20 and 40, respectively, bits per second in situations in which the pace is not limited by judgmental processes).

In computer-aided teaching, the restriction to a small ensemble of multiple-choice responses sometimes precludes truly convenient, natural communication, and it leads into controversy with those who think that the "constructed response" methods are inherently superior. In our work thus far, it appears that the difference in effectiveness between constructed-response and multiple-choice procedures is small compared with the difference between a convenient, fast response mode and an inconvenient, slow one. Convenience and speed influence markedly the student's enjoyment of his interaction with the computer and the lesson. The most important sub-goal, we believe, is to maximize the amount of enjoyment, satisfaction, and reinforcement the student derives from the interaction. And good student-teacher communication appears to be absolutely essential to that maximization.

Good man-computer communication is important, also, in systems in which the computer serves to facilitate learning without taking the initiative characteristic of most human teachers. We are working on a system, *Graph Equation*, the aim of which is to facilitate a student's exploration of the relations between the symbolic and graphical forms of mathematical equations.

The program displays, for example, the graph of a parabola (see Fig. 2), and below the graph it displays the equation,

$$y = a(x-b)^2 + c \quad . \qquad (1)$$

Associated with each of the parameters, *a*, *b*, and *c*, is a potentiometer that controls the value of the parameter. The student can vary the parameter values at will and see, directly and immediately, the correspondence between the configuration of those values and the shape of the parabola. We are in the process of substituting, for the potentiometers, "light scales" with pointers operated by the light pen and of displaying numerical coefficients instead of letter parameters on the oscilloscope. Even in the present crude form, however, the system is an effective aid. It presents the linkage between the symbolic and the graphical representation in a dynamic way. It lets the student explore many more configurations than he could explore if he had to plot graphs on paper. And it lets him see "answers" while he is still thinking about "questions" -- something we think may be very important in learning.

We plan, of course, to have the *Graph Equation* system operate dynamically in the other direction, also. The student will draw a rough parabola. The computer will fit an accurate parabola to the rough one and display the accurate one. At the same time, the computer will calculate and display the coefficients. The completed system, we hope, will provide the student with a flexible, responsive study tool. It will not have much practical value as long as it is restricted to parabolas, of course, but it should be possible, with a faster machine, to handle Fourier transforms, convolution integrals, and the like.*

Often the student must manipulate characters of text with reference to pictorial or graphical information. We have been able to handle some of these functions but still lack an integrated system for communication of interrelated symbolic and pictorial information between the student and the computer.

---

* The work on computer-aided teaching and learning is supported by the United States Air Force under Contract No. AF33(616)-8152 and is monitored by the Training Psychology Branch, Behavioral Sciences Division, Aerospace Medical Research Laboratory, Aeronautical Systems Division, Air Force Systems Command.

Computer-Aided Planning and Design

In starting to explore the field of computer-aided planning and design of systems, we have focused on hospitals. Hospitals pose very interesting and difficult -- and we believe to a large extent typical -- system problems because the relative importance of the various planning factors varies from one local context to another, because so many kinds of interest and experience are relevant and eager to make themselves felt, and because tangibles and intangibles are so intimately interrelated. One of the main aims in setting up a computer system to facilitate hospital planning is therefore to provide a means through which general guide lines and local constraints can interact. Another is to permit several persons with various backgrounds and interests to look at tentative plans from their own differing points of view and to manipulate and transform the plans during the course of their discussion. A third (since the intangible factors must ultimately be converted into tangible, physical form) is to give the planners a way of sketching out their suggestions and then relating them, quickly and conveniently, to all the other considerations that have been introduced.

Coplanner, a computer-oriented planning system with which we have been working, is essentially:

1. The PDP-1 computer with typewriter, oscilloscope, light gun, and magnetic-tape unit.
2. An ensemble of empirical data describing the commerce (communication of information, transportation of objects, and movement of personnel) that goes on in typical hospitals.
3. An ensemble of programs for accepting, storing, retrieving, processing and displaying information.

In our work thus far with *Coplanner*, we have experimented with hypothetical hospital situations, using two or three members of our own group and an outside expert or two as the planning team. In preparation for a team planning session, we load into core the programs most likely to be wanted first and make ready the tapes containing the rest of the programs, the ensemble of empirical data, and the material generated in previous planning sessions.

The members of the planning team then sit before the oscilloscope. They start to discuss, for example, a hospital that is expanding its plant and must relocate and enlarge its X-Ray Department. They come to the question: Where should the X-Ray Department be located, relative to the other departments and facilities, in order to minimize the cost of its interdepartmental commerce?

One of the members of the team retrieves, through the computer, a record of previous analyses that provides data on the major components of X-Ray commerce:

a. transport of patients,

b. trips by doctors and internes to supervise x-ray examinations, to study x-ray films, and to consult with personnel of the X-Ray Department,

c. communication not involving movement of personnel, and

d. routine personnel activities such as entering or leaving duty stations and taking meals and breaks.

In response to typewriter commands, the computer then prepares and displays several graphs to summarize the quantitative commerce data. The graphs are mainly distribution graphs and histograms. Since they refer to hospitals of the same type and size, but not to precisely the one being planned, intuitive judgment suggests modifications to take into account various features of the local context. Members of the team make the adjustments in the process of discussion. All they have to do to increase the height of a bar in a histogram is to touch the top of the bar with the light pen and lift it to the desired level. Usually there will be discussion of the change and several successive adjustments of the graph. If the graph is a frequency histogram, raising one bar automatically lowers the others. Efforts have been made to create a favorable context for exercise of the planners' intuitive judgment. Provision is made for labeling, filing, and later processing alternative quantitative summaries if the planners do not agree fully on a single summary.

Figure 3 shows two graphs of the type developed in this phase of the planning discussion.

The planners of course have several different ideas concerning the new layout of the hospital. To make these ideas concrete, they display prepared floor plans -- a separate plan for each floor of each version -- or sketch them directly on the screen of the oscilloscope, using the light pen as a stylus. Sketching is facilitated by the computer, which posts a background outline plan having the proper dimensions and showing existing structures that cannot readily be altered. In its "straight-line" mode, the computer plots straight lines even if the sketcher's lines are wavy. In its "preferentially-parallel-to-axes" mode, the computer plots lines precisely parallel to the x axis if the sketcher makes them approximately so, etc. On their sketches, which they can readily file away and recall for revision, the planners label the various departments and the stairs, elevators, dumbwaiters, etc. Each label, typed on the typewriter, appears at the top of the oscilloscope screen, and then is adjusted to desired size, trapped by the light pen, moved to its proper location on the plan, and dropped there. Each label serves as a storage and retrieval tag for the sketch to which it is attached. The plan can therefore be made up in small parts and displayed as a whole. Within a few months, the program will be capable of filing and retrieving assemblies by name.

Having tentatively worked out their ideas about X-Ray commerce and sketched several physical arrangements, the planners now turn to the problem of evaluation. First, they select one of the commerce-distribution hypotheses and one of the physical layouts for examination and designate them as input data to a fast-time simulation program that converts the commerce pattern from a set of statistical distributions to a sequence of individual trips and calls. Then they apply a program that finds the best routes for the trips and calls and computes expected durations and costs. In calculating cost, the amounts of time spent by various categories of personnel are weighted appropriately. The weighting function can, of course, be discussed and varied by the planning team. The calculated cost provides an evaluative measure for the selected layout under the selected commerce hypothesis. Actually, several different evaluative formulas are ordinarily used. The corresponding cost figures are saved for later use.

The evaluative procedure is then applied to other combinations of layout and commerce hypothesis. When all the combinations have been treated, the planners recall the cost figures and compare them. On the basis of this comparison, they usually discard all but the best two or three schemes. They modify the best ones, introduce new considerations developed as a result of the study, and make further simulation and evaluation runs.

Figure 5 shows an output-display prepared by the evaluation program.

If the planners are inclined to go into detail in certain areas, Coplanner is prepared to assist them. An elevator-simulation routine, for example, provides a dynamic display of elevator operation under the loads specified by a selected commerce-distribution hypothesis and a determination of best routes. Direct dynamic simulation has important roles to play in work of this kind because it appeals to non-mathematical planners more directly than does queuing-theory analysis performed with the aid only of symbolic assumptions and equations. Sometimes dynamic simulation is a substitute for the abstract theory; sometimes it is an introduction to the abstract theory; sometimes it is a check upon the abstract theory.

In the preceding discussion, one small facet of the hospital planning problem was used to illustrate the approach we are advocating. We have developed a fairly powerful system to facilitate planning in the area discussed and in related areas. In other areas, the system is only starting to develop. The computer parts of the system are not intended, we should emphasize, to calculate optimal plans or designs; they are intended to provide memory, manipulative, computing, and display functions in such a way that they can be integrated with the more intuitive functions supplied by the human parts of the system.*

Visualizing the Operation of Computer Programs

The covertness of the operation of the programs of electronic computers makes it difficult for us to develop of them the same direct, perceptual kind of comprehension that most of us have of familiar mechanisms, the moving parts of which we can see and touch. The great speed with which the programs run

adds to the difficulty, of course, but we are in the habit of solving the speed problem -- for example, through "slow motion." Unless a window or a plastic model will provide solution, however, we are in the habit of letting the problem of covertness go unsolved. We tend to be satisfied with extremely indirect procedures for interrogation and for drawing inferences. In the case of the human brain, for example, a neurophysiologist may try to construct a model of an internal process on the basis of waveforms recorded from 10 or 100 of the million or billion neurons involved, plus microscopic inspection of several slices of the tissue prepared in such a way as to render visible one or another feature of its architecture. Our approach to computers is comparable: When trouble arises and the results do not turn out as we expect them to, we may try to figure out what is going on by examining with the aid of a typewriter control program the contents of supposedly critical registers, one register at a time, even though we cannot hope to look at more than a hundred of the thousands or tens of thousands of registers involved. Alternatively, we may ask for a printout of the contents of many registers at some particular point in the running of the program, hoping to reconstruct the dynamic pattern of events from the static view provided by the printout.

Considering the problem posed by covertness leads one to think about the procedure, introspection, used as the basic experimental tool in such early psychological laboratories as Wundt's and Titchener's, and still widely employed in the development, if not in the formal testing, of psychological hypotheses. Human introspection is a useful procedure despite its severe shortcomings. How much more useful it would be if those shortcomings were overcome -- if all the processes of the brain were accessible to the reporting mechanism; if the reporting mechanism could describe all the aspects of those processes; if the reports were detailed and accurate; if introspecting did not interfere with the process under examination.

That thought leads immediately to the idea of a computer analogy to, or improvement upon, human introspection. Clearly, computer introspection can be freed of all the shortcomings mentioned, except the last, and the last one can be turned to advantage. Displaying its own internal processes will of course interfere with the computer's execution of its substantive programs, but only by appropriating memory space and time. Often, there is memory space to spare, and programs normally run too fast for the operator to follow them perceptually. The conclusion, therefore, is that it might be interesting to experiment with programs that display various aspects of the internal operation of the running computer.

Two such programs, written for the PDP-1 computer, are Program Graph and Memory Course. Program Graph was written with the hope that it would facilitate the introduction to computer programming and provide displays through which certain individual or "personality" characteristics of programming style may be seen. Memory Course was intended mainly for use in "debugging" computer programs. Both programs make use of a trace routine that executes the instructions of the object program in normal, running sequence and, after each execution, (a) records in core registers the contents of the accumulator, input-output register, and program counter, (b) does some incidental bookkeeping, and (c) turns control over to the display routines. The display routines develop graphs of types to be illustrated.

The graphs displayed by Program Graph are illustrated in Fig. 6. In Fig. 6A, as each instruction of the object program is executed, its location is plotted as ordinate, and the cumulative number of executions is plotted as abscissa. (Roughly speaking, therefore, the graph represents active memory location versus time.) Both the ordinate and the abscissa scales run from 0 to 1777 (octal). The interpretation of the graph is quite direct: straight-line parts of the graph represent straight-line parts of the program; jumps represent jumps or subroutine calls; serrations represent loops. The subroutine structure is revealed clearly. If the operator knows the general course the program should follow, he can detect and locate gross faults readily.

Figures 6B-6D show, for the same object program, the contents of the accumulator as a function of time. The abscissa scale again runs from 0 to 1777. In Fig. 6B, the ordinate scale covers the range from -2^{17} to 2^{17}; in Fig. 6C, it runs from -2^{13} to 2^{13}; and in Fig. 6D, it runs from -2^{7} to 2^{7}. Evidently, the accumulator is heavily engaged in computations involving small numbers.

---

*Coplanner was developed under USPHS Project W-59, Collaborative Research in Hospital Planning. J.J. Souter, A.I.A., and M.B. Brown, M.D., past and present Project Directors, and J.I. Elkind and W.E. Fletcher participated in the formulation of the system.

Figures 6E-6G show, for the input-output register, what Figs. 6B-6D showed for the accumulator.

Figure 6H displays the instruction codes. Each instruction code is a two-digit octal number. The ordinate scale extends from 02 (*and*) to 76 (*operate*, which is an augmented instruction, the augmentation not shown). The most heavily used instructions are 20 (*load accumulator with contents of*) and 24 (*deposit contents of accumulator into*).

Figure 6I displays the memory references and the augmentations. Both are shown here; either class may be suppressed.

In Fig. 6J, all the graphs of Figs. 6A-6I are displayed simultaneously. Because the points are shape-coded, it is possible, though difficult, to reconstruct in detail the sequential pattern of a program from graphs of this type. They might therefore find application in historical documentation of very critical computations, such as those concerned with rocket launching and air defense. In any event, the composite representation conveys an impression of the great capability computer's have to introspect upon their internal processes and report about them in detail.

As we leave this topic, we should perhaps mention the phenomenon that appears when *Program Graph* is equipped for recursive operation and set to display its own operation. The result, of course, is only a recursion of beginnings, terminated by overflowing of the pushdown list. This effect is not entirely foreign to human introspection.

The routine, *Memory Course*, plots a grid-like map of memory and displays, against the background of the grid, the course through memory taken by the object program. The dots of the grid represent memory registers, and the dot that represents the register containing the instruction presently being executed is encircled. As control passes from one instruction to another of the object program, a line is drawn connecting the corresponding registers. The effect is hard to illustrate in a still photograph because its effectiveness depends largely upon the kinetic character of the display. However, Fig. 7 may convey an approximate impression. Because the photograph integrates over time, it shows a longer segment of the program's course through memory than one sees when he views the oscilloscope directly.

Program Graph and *Memory Course* are but two of many possible schemes for displaying the internal processes of the computer. We are working on others that combine graphical presentation with symbolic presentation. Symbolic presentation is widely used, of course, in "debugging" routines. If many symbols are displayed, however, it is not possible to proceed through the program rapidly enough to find errors in reasonable time. By combining graphical with symbolic presentation, and putting the mode of combination under the operator's control via light pen, we hope to achieve both good speed and good discrimination of detailed information.*

Problems to be Solved in Man-Computer Communication

Among the problems toward which man-computer symbiosis is aimed -- problems that men and computers should attack in partnership -- are some of great intellectual depth and intrinsic difficulty. The main problems that must be solved to bring man-computer symbiosis into being, however, appear not to be of that kind. They are not easy, but their difficulty seems due more to limitations of technology than to limitations of intelligence.

What we would like to achieve, at least as a sub-goal, is a mechanism that will couple man to computer as closely as man is now coupled to man in good multidisciplinary scientific or engineering teams.

For a psychologist to telephone a mathematician and ask him, "How can I integrate $\int(dx/(1-x^2))$?" required, in one empirical test, 105 seconds, including 65 seconds devoted to dialing and formalities with the mathematician's secretary plus 32 seconds of preamble with the mathematician. To ask the mathematician that particular question is, of course, wantonly to waste his time -- 170 seconds of it, in this case, since all he needed to say was: "Look it up in any table of integrals," and all he did say was that sentence embedded in a context of encouragement and courtesy. (To find a table of integrals[2] and then to locate the entry took the psychologist, who missed the relevant formula on his first pass and started over at the beginning after scanning 569 entries, 7 minutes and 25 seconds.)

What we would like the computer to do for us, in the context of the foregoing example, does not require such a deep solution as an algorithm for formal differentiation; it requires merely good communication and retrieval. We would like to have an arrangement that would let the psychologist write on his desk input-output surface:

$$\int \frac{dx}{1-x^2} = \text{what?} \quad , \qquad (2)$$

and then let the computer replace the "what?" -- in perhaps 2 or even 20 seconds -- by the expression:

$$\frac{1}{2}\log\frac{|1+x|}{|1-x|} \quad . \qquad (3)$$

In the example, our aspiration would not stop, of course, with the display of expression (3) in symbolic form. The psychologist would surely want elucidation. His next request might be "Please plot a graph," or, if the novelty were worn off, simply "Graph." We would then like to have the computer display on the input-output surface a figure such as Fig. 140 in Dwight's *Tables*.[3] The figure would, of course, be plotted from computed points, not retrieved from storage. It would be no trouble for the computer to calculate and present it in a few seconds. (For the psychologist to plot a rough graph of the integral took 12 minutes. For another person to locate a published figure (Dwight's) took 17 minutes: a little more than 16 to get to the document room and thumb through books that did not contain the figure, and then a little less than 1 to pick up Dwight's book and scan as far as page 29, where the figure is.)

Five Immediate Problems

Consideration of many such examples as the foregoing and of what would have to be done to put the computer's clerical power conveniently and responsively under the control of human initiative suggests that the main essential steps to man-computer symbiosis are the following:

1. For the economic reason mentioned in the Introduction, develop systems for sharing the time of digital computers among many users on a split-millisecond basis. With J. McCarthy and S. Boilen, one of us is working on a small-scale prototype of such a system with five user stations.*

2. Devise an electronic input-output surface on which both the operator and the computer can display, and through which they can communicate, correlated symbolic and pictorial information. The surface should have selective persistence plus selective erasability; the computer should not have to spend a large part of its time maintaining the displays. The entire device should be inexpensive enough for incorporation into a remote console. An interesting approach to the man-to-machine part of this problem is being taken by Teagher.[4] We are employing an oscilloscope and light pen to fulfill the function, but they do not meet the cost and selective-persistence requirements.

3. Develop a programming system that will facilitate real-time contingent selection and shaping of information-processing procedures. The system must permit trial-and-error operation based upon "tentative computation": it will often be necessary to go back to the beginning or to an intermediate point and to try a different attack. We are experimenting with interpretive systems for on-line assembly of procedures from sub-procedures,** and we are planning work on console compiling, intermeshed with testing and contingent application of procedures as they are required by the human components of the man-computer partnership.

4. Develop systems for storage and retrieval of the vast quantities of information required to support, simultaneously at several user stations, creative thinking in various areas of investigation. For economic reasons, such systems must almost certainly be hierarchical, moving information from large-capacity, fast-access storage as (or shortly before) the information is needed. To achieve the desired effectiveness, it will probably be necessary to make advances in the direction of parallel-access, associative memory with preliminary activation based upon apperceptive relevance. In this area, we believe, much fundamental study of

---

* Preliminary study of these displays of internal computer processes was supported through a contract with the Council on Library Resources.

* The work on time-sharing is supported by Grant R68568 from the National Institute of Health.

**One of the systems is based on a typewriter control program, *Process Control*, written by D. Park.

information indexing and of memory organization will be necessary before truly satisfactory hardware can be designed, but it appears that quite a bit can be accomplished directly through development of memories -- probably read-only memories -- with very large capacity and moderately fast access and through the application of existing keyword or descriptor techniques.

5. Solve the problem of human cooperation in the development of large program systems. It appears that the development of effective human cooperation and the development of man-computer symbiosis are "chicken-and-egg" problems. It will take unusual human teamwork to set up a truly workable man-computer partnership, and it will take man-computer partnerships to engender and facilitate the human cooperation. For that reason, the main tasks of the first time-sharing computer system with many remote stations may well be in the areas of language and procedure development.

In the five problem areas just mentioned, "to begin is everything," even if it is necessary at first to build research systems along lines that would be uneconomic for widespread application. If we neglect the arguments of economics and elegance we can think at once of ways of solving, or at least starting to solve, the problems. These ways will probably be adequate to test the premise that man-computer symbiosis will be able to achieve intellectual results beyond the range of men alone or of computers programmed and operated in conventional ways.

Four Long-Term Problems

In four other areas, the problems to be solved appear -- if they are not simplified beyond recognition in the effort to make them tractable -- to be deep and intrinsically difficult. The first of these areas is computer appreciation of natural written languages, in their semantic and pragmatic as well as in their syntactic aspects. The second is computer recognition of words spoken in context by various and unselected talkers. The third is the theory of algorithms, particularly their discovery and simplification. The fourth is heuristic programming. We believe that these four areas will in the long term be extremely important to man-computer symbiosis, but that man-computer partnerships of considerable effectiveness and value can be achieved without them. We suspect that solutions in these areas will be found with the aid of early man-computer symbioses, rather than conversely.

An Intermediate Problem

A system combining an elementary form of computer speech recognition, computer recognition of carefully hand-printed characters, and simple light-pen editing techniques, would provide, we think, a very convenient and effective communication link between man and computer. The problems involved in creating such a system seem to us to be intermediate between the five and the four. They may be solved in time to permit the use of correlated voice-hand input in the earliest man-computer partnerships, but, if the required solutions are not ready, it would not be good to wait for them.

References

1. James J. Souder, Madison Brown, Weldon Clark and Jerome Elkind, Collaborative Research in Hospital Planning, United States Public Health Service, Project W-59, to be published in the spring of 1962.

2. B. O. Peirce, A Short Table of Integrals, 3rd edition, Boston: Ginn and Company, 1929.

3. H. B. Dwight, Table of Integrals and other Mathematical Data, 3rd edition, New York: The Macmillan Company, 1957.

4. H. M. Teager, Semi-Annual Progress Reports dated January 1961, June 1961 and January 1962, M.I.T. Computation Center. Also, Quarterly Progress Reports 2, 3, 4 and 5 of the Real-Time Time-Sharing Project, M.I.T. Computation Center.

```
0

Good afternoon.  This will be
your German-English Lesson No. 4.
If you are ready to start at
once, please type "s."  If you
would like to review the
procedure, please type "p."

s

reichen           to hand• +
reichen           to pass• +
    64                64   good
öffnen            to offer• - ta   n
to open           to open°
   -120              -56   poor
arbeiten          to arbitrate° - ta    y
arbeiten          to look° - ta   n
to work           to work•
   -184             -240   Dumbkopf!
kochen            to cook• - ta   y
kochen            to boil• +
     0              -240   okay
öffnen            to open• +
    64              -176   hot dog
rauchen           to smoke• +
    64              -112   admirable
arbeiten          to work• +
    64               -48   good
kochen            to boil• +
    64                16   very good
machen            to make• +
    64                80   Keep it up.
    80

That's it.  You did well.  I'll
be looking forward to the next
lesson.
```

Fig. 1 -- Typescript of a short illustrative lesson in which a computer plays the role of instructor in language-vocabulary drill. The student typed "O" to start the session, "s" to start the lesson, the English words (and terminating dots) in right-hand column, and the abbreviations of "yes" and "no" in response to the computer's "ta" ("Do you want to try again?"). The computer typed the remainder, including scores and comments. The procedure is explained in the text.

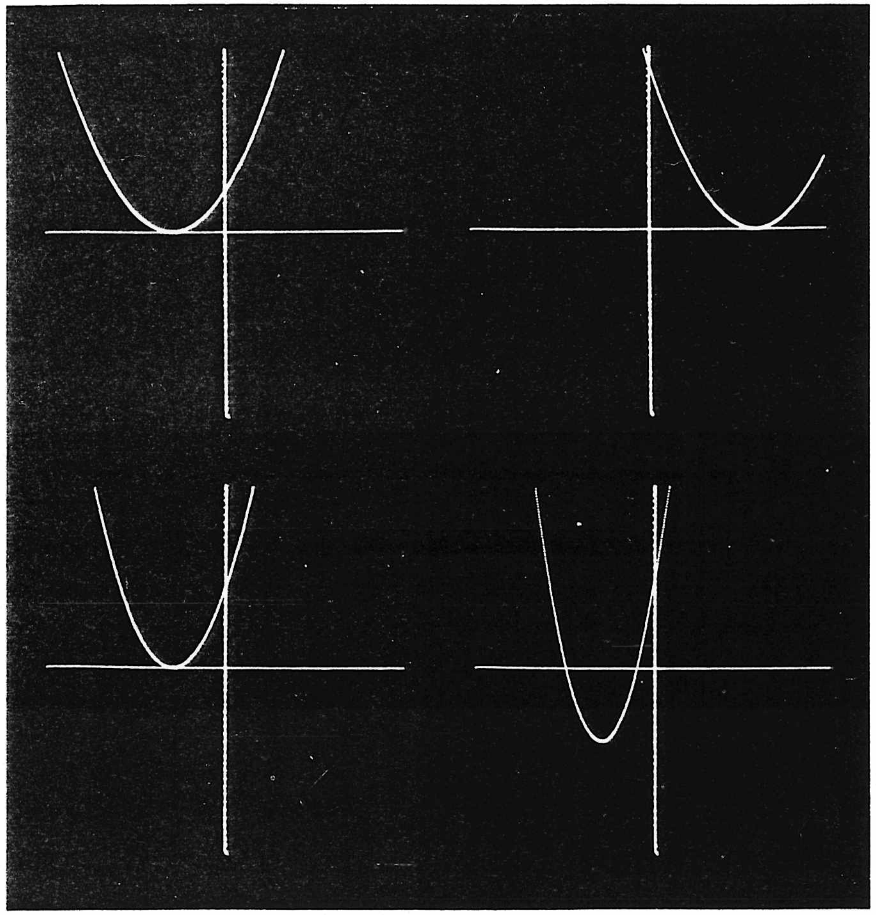

Fig. 2 -- Parabolas displayed by computer to facilitate student's exploration of relations between graphical and symbolic representations of mathematical expressions.

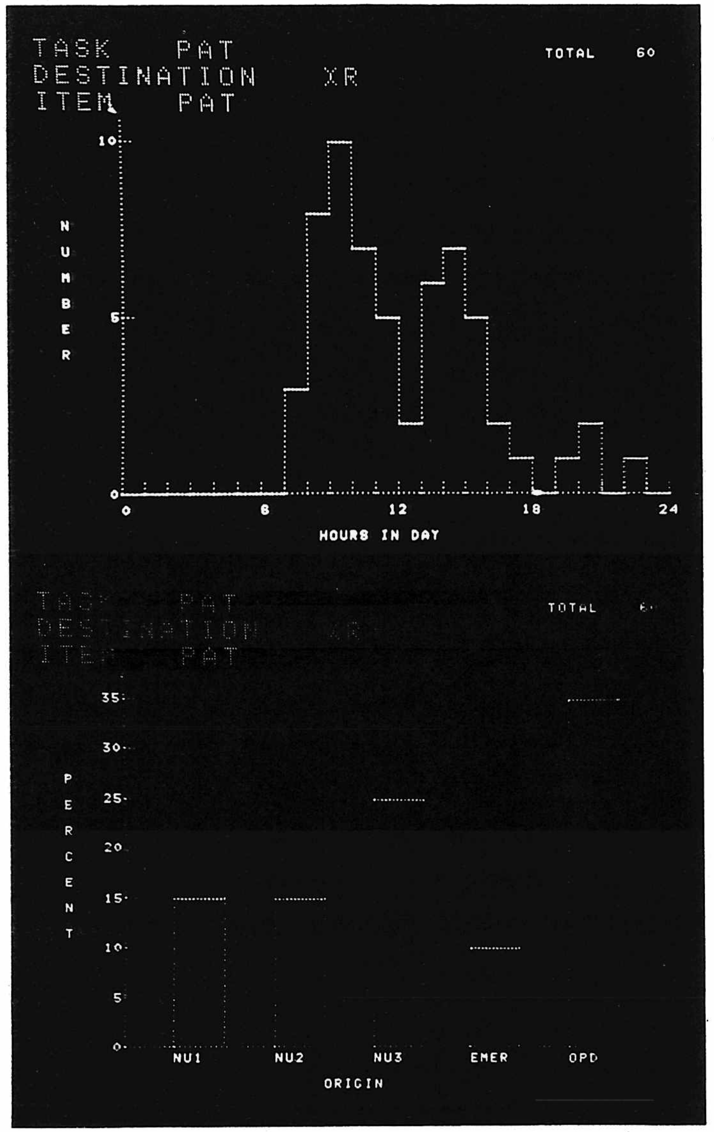

Fig. 3 -- Oscilloscope displays of several aspects of a projected inter-departmental "commerce" pattern in a hypothetical hospital. The upper graph shows the anticipated time distribution of patient transport trips to the X-Ray Department. The lower graph shows the conditional distribution of those trips among departments of origin.

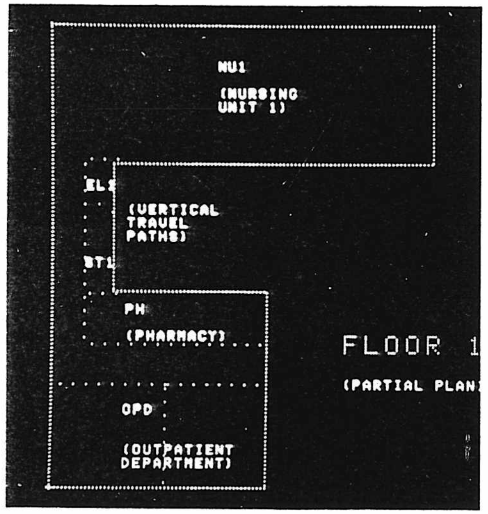

Fig. 4 -- Oscilloscope display of an outline planning sketch of one floor in a hypothetical hospital.

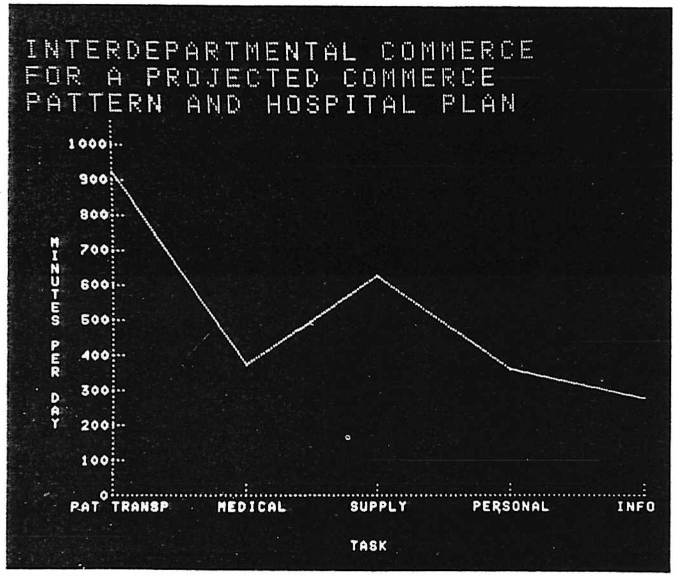

Fig. 5 -- Oscilloscope display of the performance, in respect of "commerce," of a proposed hospital plan. Scale time is defined as man-minutes spent in transit. The contributions of individuals to this quantity are weighted by coefficients associated with their personnel categories.

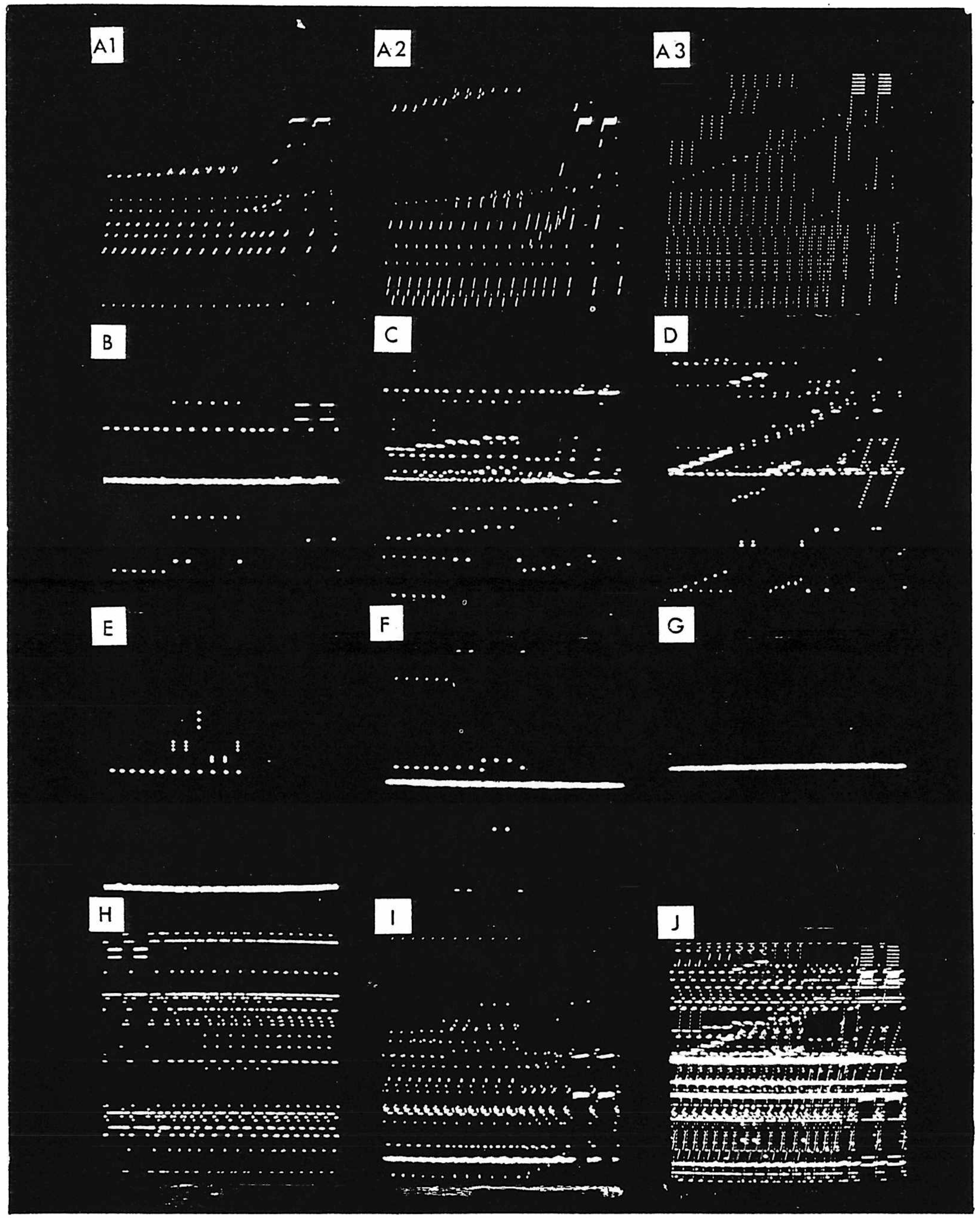

Fig. 6 -- Photographs of oscilloscopic displays made by Program Graph. See text for interpretation.