61

In Memoriam: J. C. R. Licklider
1915-1990

August 7, 1990

digital

Systems Research Center
130 Lytton Avenue
Palo Alto, California 94301

Systems Research Center

The charter of SRC is to advance both the state of knowledge and the state of the art in computer systems. From our establishment in 1984, we have performed basic and applied research to support Digital's business objectives, Our current work includes exploring distributed personal computing on multiple platforms, networking, programming technology, system modelling and management techniques, and selected applications.

Our strategy is to test the technical and practical value of our ideas by building hardware and software prototypes and using them as daily tools. Interesting systems are too complex to be evaluated solely in the abstract; extended use allows us to investigate their properties in depth. This experience is useful in the short term in refining our designs, and invaluable in the long term in advancing our knowledge. Most of the major advances in information systems have come through this strategy, including personal computing, distributed systems, and the Internet.

We also perform complementary work of a more mathematical flavor. Some of it is in established fields of theoretical computer science, such as the analysis of algorithms, computational geometry, and logics of programming. Other work explores new ground motivated by problems that arise in our systems research.

We have a strong commitment to communicating our results; exposing and testing our ideas in the research and development communities leads to improved understanding. Our research report series supplements publication in professional journals and conferences. We seek users for our prototype systems among those with whom we have common interests, and we encourage collaboration with university researchers.

Robert W. Taylor, Director

In Memoriam:
J. C. R. Licklider

1915–1990

©IRE (now IEEE) 1960
"Man-Computer Symbiosis" is reprinted, with permission, from *IRE Transactions on Human Factors in Electronics,* volume HFE-1, pages 4–11, March 1960.

©Science and Technology 1968
"The Computer as a Communication Device" is reprinted from *Science and Technology,* April 1968.

©Digital Equipment Corporation 1990
This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of the Systems Research Center of Digital Equipment Corporation in Palo Alto, California; an acknowledgment of the authors and individual contributors to the work; and all applicable portions of the copyright notice. Copying, reproducing, or republishing for any other purpose shall require a license with payment of fee to the Systems Research Center. All rights reserved.

Preface

This report honors J. C. R. Licklider for his contributions to computer science research and education in this country. We reprint here two of his papers, originally published in the 1960s, which exemplify his ideas about how computers could enhance human problem-solving.

If you were ever fortunate enough to meet him, and you said something like, "It's nice to meet you, Dr. Licklider," he would ask right away that you please call him Lick. He was Lick to friends, colleagues, and casual acquaintances alike.

Lick had a vision of a better way of computing. Once upon a time, to get a computer to do your bidding, you had to punch holes in paper cards or tapes, give the paper to someone who fed it to the machine, and then go away for hours or days. Lick believed we could do better and, more than any other single individual, saw to it that we did.

In the paper entitled "Man-Computer Symbiosis," published thirty years ago, Lick provided a guide for decades of computer research to follow. The paper was based on work performed by a small research group organized and headed by him at Bolt, Beranek, and Newman. In the late 1950s, the group purchased the first PDP- 1 from Digital. On this machine, Lick's group designed and built one of the earliest time-sharing systems.

In 1962, Lick was asked by the Director of the Advanced Research Projects Agency (ARPA) to join the agency to create and manage a program for funding research. Although its annual budget was greater than the total amount of money allocated to computer research by all other government-supported agencies, it was one of the smaller programs within ARPA. This program led the way to commercial time-sharing in the late 60s and to networking in the mid-70s.

The computer establishment criticized Lick's ARPA program. Most computer manufacturers and directors of computer centers argued that time-sharing was an inefficient use of machine resources and should not be pursued. But Lick had the courage to persevere.

His ARPA responsibilities included selecting and funding researchers to build and lead research groups. In this connection, Lick was the architect of Project MAC at MIT and a number of other projects that shaped the field. The leaders he chose twenty-five years ago now read like a Who's Who of computing research.

The least known of Lick's accomplishments is perhaps his most significant. Prior to his work at ARPA, no U.S. university granted a Ph.D. in

computer science. A university graduate program requires a research base, and that in turn requires a long-term commitment of dollars. Lick's ARPA program set the precedent for providing the research base at four of the first universities to establish graduate programs in computer science: U.C. Berkeley, CMU, MIT, and Stanford. These programs, started in 1965, have remained the country's strongest and have served as role models for other departments that followed. Their success would have been impossible without the foundation put in place by Lick in 1962-64.

For all his considerable influence on computing, Lick retained his modesty. He was the most unlikely "great man" you could ever encounter. His favorite kind of joke was one at his own expense. He was gentle, curious, and outgoing.

Lick's vision provided an extremely fruitful, long-term direction for computing research. He guided the initial research funding that was necessary to fulfil the early promises of the vision. And he laid the foundation for graduate education in the newly created field of computer science. All users of interactive computing and every company that employs computer people owe him a great debt.

Robert W. Taylor

Contents

Man-Computer Symbiosis 1
J.C.R. Licklider

The Computer as a Communication Device 21
J.C.R. Licklider and Robert W. Taylor

Man-Computer Symbiosis

Summary

Man-computer symbiosis is an expected development in cooperative interaction between men and electronic computers. It will involve very close coupling between the human and the electronic members of the partnership. The main aims are 1) to let computers facilitate formulative thinking as they now facilitate the solution of formulated problems, and 2) to enable men and computers to cooperate in making decisions and controlling complex situations without inflexible dependence on predetermined programs. In the anticipated symbiotic partnership, men will set the goals, formulate the hypotheses, determine the criteria, and perform the evaluations. Computing machines will do the routinizable work that must be done to prepare the way for insights and decisions in technical and scientific thinking. Preliminary analyses indicate that the symbiotic partnership will perform intellectual operations much more effectively than man alone can perform them. Prerequisites for the achievement of the effective, cooperative association include developments in computer time sharing, in memory components, in memory organization, in programming languages, and in input and output equipment.

1 Introduction

1.1 Symbiosis

The fig tree is pollinated only by the insect *Blastophaga grossorun.* The larva of the insect lives in the ovary of the fig tree, and there it gets its food. The tree and the insect are thus heavily interdependent: the tree cannot reproduce wit bout the insect; the insect cannot eat wit bout the tree; together, they constitute not only a viable but a productive and thriving partnership. This cooperative "living together in intimate association, or even close union, of two dissimilar organisms" is called symbiosis [27].

"Man-computer symbiosis" is a subclass of man-machine systems. There are many man-machine systems. At present, however, there are no man-computer symbioses. The purposes of this paper are to present the concept and, hopefully, to foster the development of man-computer symbiosis by analyzing some problems of interaction between men and computing machines, calling attention to applicable principles of man-machine engineering, and

pointing out a few questions to which research answers are needed. The hope is that, in not too many years, human brains and computing machines will be coupled together very tightly, and that the resulting partnership will think as no human brain has ever thought and process data in a way not approached by the information-handling machines we know today.

1.2 Between "Mechanically Extended Man" and "Artificial Intelligence"

As a concept, man-computer symbiosis is different in an important way from what North [21] has called "mechanically extended man." In the man-machine systems of the past, the human operator supplied the initiative, the direction, the integration, and the criterion. The mechanical parts of the systems were mere extensions, first of the human arm, then of the human eye. These systems certainly did not consist of "dissimilar organisms living together …" There was only one kind of organism—man—and the rest was there only to help him.

In one sense of course, any man-made system is intended to help man, to help a man or men outside the system. If we focus upon the human operator within the system, however, we see that, in some areas of technology, a fantastic change has taken place during the last few years. "Mechanical extension" has given way to replacement of men, to automation, and the men who remain are there more to help than to be helped. In some instances, particularly in large computer-centered information and control systems, the human operators are responsible mainly for functions that it proved infeasible to automate. Such systems ("humanly extended machines," North might call them) are not symbiotic systems. They are "semi-automatic" systems, systems that started out to be fully automatic but fell short of the goal.

Man-computer symbiosis is probably not the ultimate paradigm for complex technological systems. It seems entirely possible that, in due course, electronic or chemical "machines" will outdo the human brain in most of the functions we now consider exclusively within its province. Even now, Gelernter's IBM-704 program for proving theorems in plane geometry proceeds at about the same pace as Brooklyn high school students, and makes similar errors.[12] There are, in fact, several theorem-proving, problem-solving, chess-playing, and pattern-recognizing programs (too many for complete reference [1, 2, 5, 8, 11, 13, 17, 18, 19, 22, 23, 25]) capable of rivaling human intellectual performance in restricted areas; and Newell, Simon, and Shaw's

[20] “general problem solver” may remove some of the restrictions. In short, it seems worthwhile to avoid argument with (other) enthusiasts for artificial intelligence by conceding dominance in the distant future of cerebration to machines alone. There will nevertheless be a fairly long interim during which the main intellectual advances will be made by men and computers working together in intimate association. A multidisciplinary study group, examining future research and development problems of the Air Force, estimated that it would be 1980 before developments in artificial intelligence make it possible for machines alone to do much thinking or problem solving of military significance. That would leave, say, five years to develop man-computer symbiosis and 15 years to use it. The 15 may be 10 or 500, but those years should be intellectually the most creative and exciting in the history of mankind.

2 Aims of Man-Computer Symbiosis

Present-day computers are designed primarily to solve preformulated problems or to process data according to predetermined procedures. The course of the computation may be conditional upon results obtained during the computation, but all the alternatives must be foreseen in advance. (If an unforeseen alternative arises, the whole process comes to a halt and awaits the necessary extension of the program.) The requirement for preformulation or predetermination is sometimes no great disadvantage. It is often said that programming for a computing machine forces one to think clearly, that it disciplines the thought process. If the user can think his problem through in advance, symbiotic association with a computing machine is not necessary.

However, many problems that can be thought through in advance are very difficult to think through in advance. They would be easier to solve, and they could be solved faster, through an intuitively guided trial-and-error procedure in which the computer cooperated, turning up flaws in the reasoning or revealing unexpected turns in the solution. Other problems simply cannot be formulated without computing-machine aid. Poincaré anticipated the frustration of an important group of would-be computer users when he said, “The question is not, ‘What is the answer?’ The question is, ‘What is the question?’ “ One of the main aims of man-computer symbiosis is to bring the computing machine effectively into the formulative parts of technical problems.

The other main aim is closely related. It is to bring computing machines effectively into processes of thinking that must go on in "real time," time that moves too fast to permit using computers in conventional ways. Imagine trying, for example, to direct a battle with the aid of a computer on such a schedule as this. You formulate your problem today. Tomorrow you spend with a programmer. Next week the computer devotes 5 minutes to assembling your program and 47 seconds to calculating the answer to your problem. You get a sheet of paper 20 feet long, full of numbers that, instead of providing a final solution, only suggest a tactic that should be explored by simulation. Obviously, the battle would be over before the second step in its planning was begun. To think in interaction with a computer in the same way that you think with a colleague whose competence supplements your own will require much tighter coupling between man and machine than is suggested by the example and than is possible today.

3 Need for Computer Participation in Formulative and Real-Time Thinking

The preceding paragraphs tacitly made the assumption that, if they could be introduced effectively into the thought process, the functions that can be performed by data-processing machines would improve or facilitate thinking and problem solving in an important way. That assumption may require justification.

3.1 A Preliminary and Informal Time-and-Motion Analysis of Technical Thinking

Despite the fact that there is a voluminous literature on thinking and problem solving, including intensive case-history studies of the process of invention, I could find nothing comparable to a time-and-motion-study analysis of the mental work of a person engaged in a scientific or technical enterprise. In the spring and summer of 1957, therefore, I tried to keep track of what one moderately technical person actually did during the hours he regarded as devoted to work. Although I was aware of the inadequacy of the sampling, I served as my own subject.

It soon became apparent that the main thing I did was to keep records, and the project would have become an infinite regress if the keeping of records had been carried through in the detail envisaged in the initial plan.

It was not. Nevertheless, I obtained a picture of my activities that gave me pause. Perhaps my spectrum is not typical—I hope it is not, but I fear it is.

About 85 per cent of my "thinking" time was spent getting into a position to think, to make a decision, to learn something I needed to know. Much more time went into finding or obtaining information than into digesting it. Hours went into the plotting of graphs, and other hours into instructing an assistant how to plot. When the graphs were finished, the relations were obvious at once, but the plotting had to be done in order to make them so. At one point, it was necessary to compare six experimental determinations of a function relating speech-intelligibility to speech-to-noise ratio. No two experimenters had used the same definition or measure of speech-to-noise ratio. Several hours of calculating were required to get the data into comparable form. When they were in comparable form, it took only a few seconds to determine what I needed to know.

Throughout the period I examined, in short, my "thinking" time was devoted mainly to activities that were essentially clerical or mechanical: searching, calculating, plotting, transforming, determining the logical or dynamic consequences of a set of assumptions or hypotheses, preparing the way for a decision or an insight. Moreover, my choices of what to attempt and what not to attempt were determined to an embarrassingly great extent by considerations of clerical feasibility, not intellectual capability.

The main suggestion conveyed by the findings just described is that the operations that fill most of the time allegedly devoted to technical thinking are operations that can be performed more effectively by machines than by men. Severe problems are posed by the fact that these operations have to be performed upon diverse variables and in unforeseen and continually changing sequences. If those problems can be solved in such a way as to create a symbiotic relation between a man and a fast information-retrieval and data-processing machine, however, it seems evident that the cooperative interaction would greatly improve the thinking process.

It may be appropriate to acknowledge, at this point, that we are using the term "computer" to cover a wide class of calculating, data-processing, and information-storage-and-retrieval machines. The capabilities of machines in this class are increasing almost daily. It is therefore hazardous to make general statements about capabilities of the class. Perhaps it is equally hazardous to make general statements about the capabilities of men. Nevertheless, certain genotypic differences in capability between men and computers do stand out, and they have a bearing on the nature of possible man-computer symbiosis and the potential value of achieving it.

As has been said in various ways, men are noisy, narrow-band devices, but their nervous systems have very many parallel and simultaneously active channels. Relative to men, computing machines are very fast and very accurate, but they are constrained to perform only one or a few elementary operations at a time. Men are flexible, capable of "programming themselves contingently" on the basis of newly received information. Computing machines are single-minded, constrained by their " pre-programming." Men naturally speak redundant languages organized around unitary objects and coherent actions and employing 20 to 60 elementary symbols. Computers "naturally" speak nonredundant languages, usually with only two elementary symbols and no inherent appreciation either of unitary objects or of coherent actions.

To be rigorously correct, those characterizations would have to include many qualifiers. Nevertheless, the picture of dissimilarity (and therefore potential supplementation) that they present is essentially valid. Computing machines can do readily, well, and rapidly many things that are difficult or impossible for man, and men can do readily and well, though not rapidly, many things that are difficult or impossible for computers. That suggests that a symbiotic cooperation, if successful in integrating the positive characteristics of men and computers, would be of great value. The differences in speed and in language, of course, pose difficulties that must be overcome.

4 Separable Functions of Men and Computers in the Anticipated Symbiotic Association

It seems likely that the contributions of human operators and equipment will blend together so completely in many operations that it will be difficult to separate them neatly in analysis. That would be the case if, in gathering data on which to base a decision, for example, both the man and the computer came up with relevant precedents from experience and if the computer then suggested a course of action that agreed with the man's intuitive judgment. (In theorem-proving programs, computers find precedents in experience, and in the SAGE System, they suggest courses of action. The foregoing is not a far-fetched example.) In other operations, however, the contributions of men and equipment will be to some extent separable.

Men will set the goals and supply the motivations, of course, at least in the early years. They will formulate hypotheses. They will ask questions. They will think of mechanisms, procedures, and models. They will remem-

ber that such-and-such a person did some possibly relevant work on a topic of interest back in 1947, or at any rate shortly after World War II, and they will have an idea in what journals it might have been published. In general, they will make approximate and fallible, but leading, contributions, and they will define criteria and serve as evaluators, judging the contributions of the equipment and guiding the general line of thought.

In addition, men will handle the very-low-probability situations when such situations do actually arise. (In current man-machine systems, that is one of the human operator's most important functions. The sum of the probabilities of very-low-probability alternatives is often much too large to neglect.) Men will fill in the gaps, either in the problem solution or in the computer program, when the computer has no mode or routine that is applicable in a particular circumstance.

The information-processing equipment, for its part, will convert hypotheses into testable models and then test the models against data (which the human operator may designate roughly and identify as relevant when the computer presents them for his approval). The equipment will answer questions. It will simulate the mechanisms and models, carry out the procedures, and display the results to the operator. It will transform data, plot graphs ("cutting the cake" in whatever way the human operator specifies, or in several alternative ways if the human operator is not sure what he wants). The equipment will interpolate, extrapolate, and transform. It will convert static equations or logical statements into dynamic models so the human operator can examine their behavior. In general, it will carry out the routinizable, clerical operations that fill the intervals between decisions.

In addition, the computer will serve as a statistical-inference, decision-theory, or game-theory machine to make elementary evaluations of suggested courses of action whenever there is enough basis to support a formal statistical analysis. Finally, it will do as much diagnosis, pattern-matching, and relevance-recognizing as it profitably can, but it will accept a clearly secondary status in those areas.

5 Prerequisites for Realization of Man-Computer Symbiosis

The data-processing equipment tacitly postulated in the preceding section is not available. The computer programs have not been written. There are in fact several hurdles that stand between the nonsymbiotic present and the

anticipated symbiotic future. Let us examine some of them to see more clearly what is needed and what the chances are of achieving it.

5.1 Speed Mismatch Between Men and Computers

Any present-day large-scale computer is too fast and too costly for real-time cooperative thinking with one man. Clearly, for the sake of efficiency and economy, the computer must divide its time among many users. Time-sharing systems are currently under active development. There are even arrangements to keep users from "clobbering" anything but their own personal programs.

It seems reasonable to envision, for a time 10 or 15 years hence, a "thinking center" that will incorporate the functions of present-day libraries together with anticipated advances in information storage and retrieval and the symbiotic functions suggested earlier in this paper. The picture readily enlarges itself into a network of such centers, connected to one another by wide-band communication lines and to individual users by leased-wire services. In such a system, the speed of the computers would be balanced, and the cost of the gigantic memories and the sophisticated programs would be divided by the number of users.

5.2 Memory Hardware Requirements

When we start to think of storing any appreciable fraction of a technical literature in computer memory, we run into billions of bits and, unless things change markedly, billions of dollars.

The first thing to face is that we shall not store all the technical and scientific papers in computer memory. We may store the parts that can be summarized most succinctly—the quantitative parts and the reference citations—but not the whole. Books are among the most beautifully engineered, and human-engineered, components in existence, and they will continue to be functionally important within the context of man-computer symbiosis. (Hopefully, the computer will expedite the finding, delivering, and returning of books.)

The second point is that a very important section of memory will be permanent: part indelible *memory* and part *published memory.* The computer will be able to write once into indelible memory, and then read back indefinitely, but the computer will not be able to erase indelible memory. (It may also over-write, turning all the 0's into l's, as though marking over

what was written earlier.) Published memory will be "read-only" memory. It will be introduced into the computer already structured. The computer will be able to refer to it repeatedly, but not to change it. These types of memory will become more and more important as computers grow larger. They can be made more compact than core, thin-film, or even tape memory, and they will be much less expensive. The main engineering problems will concern selection circuitry.

In so far as other aspects of memory requirement are concerned, we may count upon the continuing development of ordinary scientific and business computing machines There is some prospect that memory elements will become as fast as processing (logic) elements. That development would have a revolutionary effect upon the design of computers.

5.3 Memory Organization Requirements

Implicit in the idea of man-computer symbiosis are the requirements that information be retrievable both by name and by pattern and that it be accessible through procedure much faster than serial search. At least half of the problem of memory organization appears to reside in the storage procedure. Most of the remainder seems to be wrapped up in the problem of pattern recognition within the storage mechanism or medium. Detailed discussion of these problems is beyond the present scope. However, a brief outline of one promising idea, "trie memory," may serve to indicate the general nature of anticipated developments.

Trie memory is so called by its originator, Fredkin [10], because it is designed to facilitate retrieval of information and because the branching storage structure, when developed, resembles a tree. Most common memory systems store functions of arguments at locations designated by the arguments. (In one sense, they do not store the arguments at all. In another and more realistic sense, they store all the possible arguments in the framework structure of the memory.) The trie memory system, on the other hand, stores both the functions and the arguments. The argument is introduced into the memory first, one character at a time, starting at a standard initial register. Each argument register has one cell for each character of the ensemble (e.g., two for information encoded in binary form) and each character cell has within it storage space for the address of the next register. The argument is stored by writing a series of addresses, each one of which tells where to find the next. At the end of the argument is a special "end-of-argument" marker. Then follow directions to the function, which is

stored in one or another of several ways, either further trie structure or "list structure" often being most effective.

The trie memory scheme is inefficient for small memories, but it becomes increasingly efficient in using available storage space as memory size increases. The attractive features of the scheme are these: 1) The retrieval process is extremely simple. Given the argument, enter the standard initial register with the first character, and pick up the address of the second. Then go to the second register, and pick up the address of the third, etc. 2) If two arguments have initial characters in common, they use the same storage space for those characters. 3) The lengths of the arguments need not be the same, and need not be specified in advance. 4) No room in storage is reserved for or used by any argument until it is actually stored. The trie structure is created as the items are introduced into the memory. 5) A function can be used as an argument for another function, and that function as an argument for the next. Thus, for example, by entering with the argument, "matrix multiplication," one might retrieve the entire program for performing a matrix multiplication on the computer. 6) By examining the storage at a given level, one can determine what thus-far similar items have been stored. For example, if there is no citation for Egan, J. P., it is but a step or two backward to pick up the trail of Egan, James . . . .

The properties just described do not include all the desired ones, but they bring computer storage into resonance with human operators and their predilection to designate things by naming or pointing.

5.4 The Language Problem

The basic dissimilarity between human languages and computer languages may be the most serious obstacle to true symbiosis. It is reassuring, however, to note what great strides have already been made, through interpretive programs and particularly through assembly or compiling programs such as FORTRAN, to adapt computers to human language forms. The "Information Processing Language" of Shaw, Newell, Simon, and Ellis [24] represents another line of rapprochement. And, in ALGOL and related systems, men are proving their flexibility by adopting standard formulas of representation and expression that are readily translatable into machine language.

For the purposes of real-time cooperation between men and computers, it will be necessary, however, to make use of an additional and rather different principle of communication and control. The idea may be high-lighted by comparing instructions ordinarily addressed to intelligent human beings with

instructions ordinarily used with computers. The latter specify precisely the individual steps to take and the sequence in which to take them. The former present or imply something about incentive or motivation, and they supply a criterion by which the human executor of the instructions will know when he has accomplished his task. In short: instructions directed to computers specify courses; instructions-directed to human beings specify goals.

Men appear to think more naturally and easily in terms of goals than in terms of courses. True, they usually know something about directions in which to travel or lines along which to work, but few start out with precisely formulated itineraries. Who, for example, would depart from Boston for Los Angeles with a detailed specification of the route? Instead, to paraphrase Wiener, men bound for Los Angeles try continually to decrease the amount by which they are not yet in the smog.

Computer instruction through specification of goals is being approached along two paths. The first involves problem-solving, hill-climbing, self-organizing programs. The second involves real-time concatenation of pre-programmed segments and closed subroutines which the human operator can designate and call into action simply by name.

Along the first of these paths, there has been promising exploratory work. It is clear that, working within the loose constraints of predetermined strategies, computers will in due course be able to devise and simplify their own procedures for achieving stated goals. Thus far, the achievements have not been substantively important; they have constituted only "demonstration in principle." Nevertheless, the implications are far-reaching.

Although the second path is simpler and apparently capable of earlier realization, it has been relatively neglected. Fredkin's trie memory provides a promising paradigm. We may in due course see a serious effort to develop computer programs that can be connected together like the words and phrases of speech to do whatever computation or control is required at the moment. The consideration that holds back such an effort, apparently, is that the effort would produce nothing that would be of great value in the context of existing computers. It would be unrewarding to develop the language before there are any computing machines capable of responding meaningfully to it.

5.5 Input and Output Equipment

The department of data processing that seems least advanced, in so far as the requirements of man-computer symbiosis are concerned, is the one that deals

with input and output equipment or, as it is seen from the human operator's point of view, displays and controls. Immediately after saying that, it is essential to make qualifying comments, because the engineering of equipment for high-speed introduction and extraction of information has been excellent, and because some very sophisticated display and control techniques have been developed in such research laboratories as the Lincoln Laboratory. By and large, in generally available computers, however, there is almost no provision for any more effective, immediate man-machine communication than can be achieved with an electric typewriter.

Displays seem to be in a somewhat better state than controls. Many computers plot graphs on oscilloscope screens, and a few take advantage of the remarkable capabilities, graphical and symbolic, of the charactron display tube. Nowhere, to my knowledge, however, is there anything approaching the flexibility and convenience of the pencil and doodle pad or the chalk and blackboard used by men in technical discussion.

1) *Desk-Surface Display and Control:* Certainly, for effective man-computer interaction, it will be necessary for the man and the computer to draw graphs and pictures and to write notes and equations to each other on the same display surface. The man should be able to present a function to the computer, in a rough but rapid fashion, by drawing a graph. The computer should read the man's writing, perhaps on the condition that it be in clear block capitals, and it should immediately post, at the location of each hand-drawn symbol, the corresponding character as interpreted and put into precise type-face. With such an input-output device, the operator would quickly learn to write or print in a manner legible to the machine. He could compose instructions and subroutines, set them into proper format, and check them over before introducing them finally into the computer's main memory. He could even define new symbols, as Gilmore and Savell [14] have done at the Lincoln Laboratory, and present them directly to the computer. He could sketch out the format of a table roughly and let the computer shape it up with precision. He could correct the computer's data, instruct the machine via flow diagrams, and in general interact with it very much as he would with another engineer, except that the "other engineer" would be a precise draftsman, a lightning calculator, a mnemonic wizard, and many other valuable partners all in one.

2) *Computer-Posted Wall Display:* In some technological systems, several men share responsibility for controlling vehicles whose behaviors interact. Some information must be presented simultaneously to all the men, preferably on a common grid, to coordinate their actions. Other informa-

tion is of relevance only to one or two operators. There would be only a confusion of uninterpretable clutter if all the information were presented on one display to all of them. The information must be posted by a computer, since manual plotting is too slow to keep it up to date.

The problem just outlined is even now a critical one, and it seems certain to become more and more critical as time goes by. Several designers are convinced that displays with the desired characteristics can be constructed with the aid of flashing lights and time-sharing viewing screens based on the light-valve principle.

The large display should be supplemented, according to most of those who have thought about the problem, by individual display-control units. The latter would permit the operators to modify the wall display without leaving their locations. For some purposes, it would be desirable for the operators to be able to communicate with the computer through the supplementary displays and perhaps even through the wall display. At least one scheme for providing such communication seems feasible.

The large wall display and its associated system are relevant, of course, to symbiotic cooperation between a computer and a team of men. Laboratory experiments have indicated repeatedly that informal, parallel arrangements of operators, coordinating their activities through reference to a large situation display, have important advantages over the arrangement, more widely used, that locates the operators at individual consoles and attempts to correlate their actions through the agency of a computer. This is one of several operator-team problems in need of careful study.

3) *Automatic Speech Production and Recognition:* How desirable and how feasible is speech communication between human operators and computing machines? That compound question is asked whenever sophisticated data-processing systems are discussed. Engineers who work and live with computers take a conservative attitude toward the desirability. Engineers who have had experience in the field of automatic speech recognition take a conservative attitude toward the feasibility. Yet there is continuing interest in the idea of talking with computing machines. In large part, the interest stems from realization that one can hardly take a military commander or a corporation president away from his work to teach him to type. If computing machines are ever to be used directly by top-level decision makers, it may be worthwhile to provide communication via the most natural means, even at considerable cost.

Preliminary analysis of his problems and time scales suggests that a corporation president would be interested in a symbiotic association with

a computer only as an avocation. Business situations usually move slowly enough that there is time for briefings and conferences. It seems reasonable, therefore, for computer specialists to be the ones who interact directly with computers in business offices.

The military commander, on the other hand, faces a greater probability of having to make critical decisions in short intervals of time. It is easy to overdramatize the notion of the ten-minute war, but it would be dangerous to count on having more than ten minutes in which to make a critical decision. As military system ground environments and control centers grow in capability and complexity, therefore, a real requirement for automatic speech production and recognition in computers seems likely to develop. Certainly, if the equipment were already developed, reliable, and available, it would be used.

In so far as feasibility is concerned, speech production poses less severe problems of a technical nature than does automatic recognition of speech sounds. A commercial electronic digital voltmeter now reads aloud its indications, digit by digit. For eight or ten years, at the Bell Telephone Laboratories, the Royal Institute of Technology (Stockholm), the Signals Research and Development Establishment (Christchurch), the Haskins Laboratory, and the Massachusetts Institute of Technology, Dunn [6], Fant [7], Lawrence [15], Cooper [3], Stevens [26], and their co-workers, have demonstrated successive generations of intelligible automatic talkers. Recent work at the Haskins Laboratory has led to the development of a digital code, suitable for use by computing machines, that makes an automatic voice utter intelligible connected discourse [16].

The feasibility of automatic speech recognition depends heavily upon the size of the vocabulary of words to be recognized and upon the diversity of talkers and accents with which it must work. Ninety-eight per cent correct recognition of naturally spoken decimal digits was demonstrated several years ago at the Bell Telephone Laboratories and at the Lincoln Laboratory [4], [9]. Togo a step up the scale of vocabulary size, we may say that an automatic recognizer of clearly spoken alpha-numerical characters can almost surely be developed now on the basis of existing knowledge. Since untrained operators can read at least as rapidly as trained ones can type, such a device would be a convenient tool in almost any computer installation.

For real-time interaction on a truly symbiotic level, however, a vocabulary of about 2000 words, e.g., 1000 words of something like basic English and 1000 technical terms, would probably be required. That constitutes a challenging problem. In the consensus of acousticians and linguists, construction

of a recognizer of 2000 words cannot be accomplished now. However, there are several organizations that would happily undertake to develop an automatic recognize for such a vocabulary on a five-year basis. They would stipulate that the speech be clear speech, dictation style, without unusual accent.

Although detailed discussion of techniques of automatic speech recognition is beyond the present scope, it is fitting to note that computing machines are playing a dominant role in the development of automatic speech recognizers. They have contributed the impetus that accounts for the present optimism, or rather for the optimism presently found in some quarters. Two or three years ago, it appeared that automatic recognition of sizeable vocabularies would not be achieved for ten or fifteen years; that it would have to await much further, gradual accumulation of knowledge of acoustic, phonetic, linguistic, and psychological processes in speech communication. Now, however, many see a prospect of accelerating the acquisition of that knowledge with the aid of computer processing of speech signals, and not a few workers have the feeling that sophisticated computer programs will be able to perform well as speech-pattern recognizes even without the aid of much substantive knowledge of speech signals and processes. Putting those two considerations together brings the estimate of the time required to achieve practically significant speech recognition down to perhaps five years, the five years just mentioned.

References

[1] A. Bernstein and M. deV. Roberts, "Computer versus chess-player," *Scientific American,* vol. 198, pp. 96-98; June, 1958.

[2] W. W. Bledsoe and I. Browning, "Pattern Recognition and Reading by Machine," presented at the Eastern Joint Computer Conf., Boston, Mass., December, 1959.

[3] F. S. Cooper, et al., "Some experiments on the perception of synthetic speech sounds," *J. Acoust. Soc. Amer.,* vol. 24, pp. 597– 606; November, 1952.

[4] K. H. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits," in W. Jackson, *Communication Theory,* Butterworths Scientific Publications, London, Eng., pp. 433-441; 1953.

[5] G. P. Dinneen, "Programming pattern recognition," *Proc. WJCC,* pp. 94-100; March, 1955.

[6] H. K. Dunn, "The calculation of vowel resonances, and an electrical vocal tract," J. *Acoust. Soc. Amer.,* vol. 22, pp.740–753; November, 1950.

[7] G. Fant, "On the Acoustics of Speech," paper presented at the Third Internatl. Congress on Acoustics, Stuttgart, Ger.; September, 1959.

[8] B. G. Farley and W. A. Clark, "Simulation of self-organizing systems by digital computers." *IRE Trans. on Information Theory,* vol. IT-4, pp.76-84; September, 1954.

[9] J. W. Forgie and C. D. Forgie, "Results obtained from a vowel recognition computer program," *J. Acoust. Soc.* Amer., vol. 31, pp. 1480-1489; November, 1959.

[10] E. Fredkin, "Trie memory," *Communications of the ACM,* Sept. 1960, pp. 490-499.

[11] R. M. Friedberg, "A learning machine: Part I," *IBM J. Res. & Dev.,* vol. 2, pp. 2–13; January, 1958.

[12] H. Gelernter, "Realization of a Geometry Theorem Proving Machine." Unesco, NS, ICIP, 1.6.6, Internatl. Conf. on Information Processing, Paris, France; June, 1959.

[13] P. C. Gilmore, "A Program for the Production of Proofs for Theorems Derivable Within the First Order Predicate Calculus from Axioms," Unesco, NS, ICIP, 1.6.14, Internatl. Conf. on Information Processing, Paris, France; June, 1959.

[14] J. T. Gilmore and R. E. Savell, "The Lincoln Writer," Lincoln Laboratory, M. I. T., Lexington, Mass., Rept. 51–8; October, 1959.

[15] W. Lawrence, et al., "Methods and Purposes of Speech Synthesis," Signals Res. and Dev. Estab., Ministry of Supply, Christ- church, Hants, England, Rept. 56/1457; March, 1956.

[16] A. M. Liberman, F. Ingemann, L. Lisker, P. Delattre, and F. S. Cooper, "Minimal rules for synthesizing speech," *J. Acoust. Soc. Amer.,* vol. 31, pp. 1490-1499; November, 1959.

[17] A. Newell, "The chess machine: an example of dealing with a complex task by adaptation," *Proc. WJCC,* pp. 101-108; March, 1955.

[18] A. Newell and J. C. Shaw, 'Programming the logic theory machine." *Proc. WJCC,* pp. 230-240; March, 1957.

[19] A. Newell, J. C. Shaw, and H. A. Simon, "Chess-playing programs and the problem of complexity," *IBM J. Res. & Dev., vol. 2,* pp. 320-33.5; October, 1958.

[20] A. Newell, H. A. Simon, and J. C. Shaw, "Report on a general problem-solving program," Unesco, NS, ICIP, 1.6.8, Internatl. Conf. on Information Processing, Paris, France; June, 1959.

[21] J. D. North, "The rational behavior of mechanically extended man", Boulton Paul Aircraft Ltd., Wolverhampton, Eng.; September, 1954.

[22] O. G. Selfridge, "Pandemonium, a paradigm for learning," *Proc. Symp. Mechanisation of Thought Processes,* Natl. Physical Lab., Teddington, Eng.; November, 1958.

[23] C. E. Shannon, "Programming a computer for playing chess," *Phil. Mag.,* vol. 41, pp. 256-75; March, 1950.

[24] J. C. Shaw, A. Newell, H. A. Simon, and T. O. Ellis, "A command structure for complex information processing," *Proc. WJCC, pp.* 119–128; May, 1958.

[25] H. Sherman, "A Quasi-Topological Method for Recognition of Line Patterns," Unesco, NS, ICIP, H.L.5, Internatl. Conf. on Information Processing, Paris, France; June, 1959.

[26] K. N. Stevens, S. Kasowski, and C. G. Fant, "Electric analog of the vocal tract," *J. Acoust. Soc. Amer.,* vol. 25, pp.734-742; July, 1953.

[27] *Webster's New International Dictionary,* 2nd ed., G. and C. Merriam Co., Springfield, Mass., p. 2555; 1958.

The Computer as a Communication Device

In a few years, men will be able to communicate more effectively through a machine than face to face.

That is a rather startling thing to say, but it is our conclusion. As if in confirmation of it, we participated a few weeks ago in a technical meeting held through a computer. In two days, the group accomplished with the aid of a computer what normally might have taken a week.

We shall talk more about the mechanics of the meeting later; it is sufficient to note here that we were all in the same room. But for all the communicating we did directly across that room, we could have been thousands of miles apart and communicated just as effectively-as people-over the distance.

Our emphasis on people is deliberate. A communications engineer thinks of communicating as transferring information from one point to another in codes and signals.

But to communicate is more than to send and to receive. Do two tape recorders communicate when they play to each other and record from each other? Not really-not in our sense. We believe that communicators have to do something nontrivial with the information they send and receive. And we believe that we are entering a technological age in which we will be able to interact with the richness of living information—not merely in the passive way that we have become accustomed to using books and libraries, but as active participants in an ongoing process, bringing something to it through our interaction with it, and not simply receiving something from it by our connection to it.

To the people who telephone an airline flight operations information service, the tape recorder that answers seems more than a passive depository. It is an often-updated model of a changing situation—a synthesis of information collected, analyzed, evaluated, and assembled to represent a situation or process in an organized way.

Still there is not much direct interaction with the airline information service; the tape recording is not changed by the customer's call. We want to emphasize something beyond its one-way transfer: the increasing significance of the jointly constructive, the mutually reinforcing aspect of communication—the part that transcends "now we both know a fact that only one of us knew before." When minds interact, new ideas emerge. We want to talk about the creative aspect of communication.

Creative, interactive communication requires a plastic or moldable medium that can be modeled, a dynamic medium in which premises will flow into consequences, and above all a common medium that can be contributed to and experimented with by all.

Such a medium is at hand—the programmed digital computer. Its presence can change the nature and value of communication even more profoundly than did the printing press and the picture tube, for, as we shall show, a well-programmed computer can provide direct access both to informational resources and to the *processes* for making use of the resources,

Communication: a comparison of models

To understand how and why the computer can have such an effect on communication, we must examine the idea of modeling-in a computer and with the aid of a computer. For modeling, we believe, is basic and central to communication. Any communication between people about the same thing is a common revelatory experience about informational models of that thing. Each model is a conceptual structure of abstractions formulated initially in the mind of one of the persons who would communicate, and if the concepts in the mind of one would-be communicator are very different from those in the mind of another, there is no common model and no communication.

By far the most numerous, most sophisticated, and most important models are those that reside in men's minds, In richness, plasticity, facility, and economy, the mental model has no peer, but, in other respects, it has shortcomings. It will not stand still for careful study. It cannot be made to repeat a run. No one knows just how it works. It serves its owner's hopes more faithfully than it serves reason. It has access only to the information stored in one man's head. It can be observed and manipulated only by one person.

Society rightly distrusts the modeling done by a single mind. Society demands consensus, agreement, at least majority. Fundamentally, this amounts to the requirement that individual models be compared and brought into some degree of accord. The requirement is for communication, which we now define concisely as "cooperative modeling" —cooperation in the construction, maintenance, and use of a model.

How can we be sure that we are modeling cooperatively, that we are communicating, unless we can compare models?

When people communicate face to face, they externalize their models so they can be sure they are talking about the same thing. Even such a simple externalized model as a flow diagram or an outline-because it

can be seen by all the communicators—serves as a focus for discussion. It changes the nature of communication: When communicators have no such common framework, they merely make speeches *at* each other; but when they have a manipulable model before them, they utter a few words, point, sketch, nod, or object.

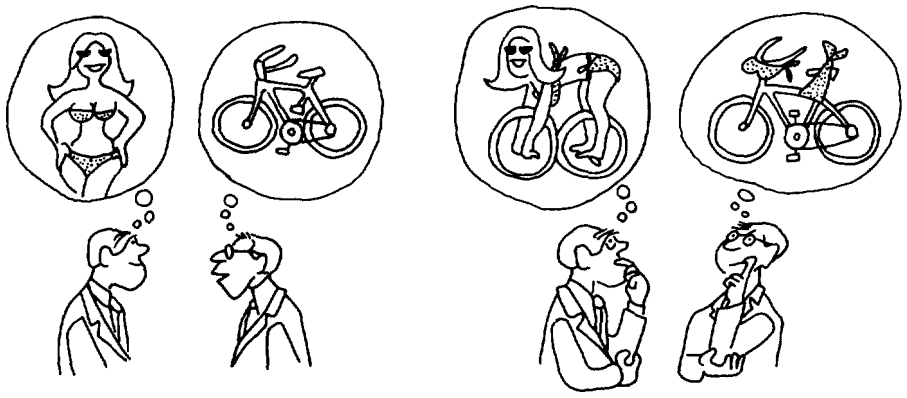
The dynamics of such communication are so model-centered as to suggest an important conclusion: Perhaps the reason present-day two-way telecommunication falls so far short of face-to-face communication is simply that it fails to provide facilities for externalizing models. Is it really seeing the expression in the other's eye that makes the face-to-face conference so much more productive than the telephone conference call, or is it being able to create and modify external models?

The project meeting as a model

In a technical project meeting, one can see going on, in fairly clear relief, the modeling process that we contend constitutes communication. Nearly every reader can recall a meeting held during the formulative phase of a project. Each member of the project brings to such a meeting a somewhat different mental model of the common undertaking—its purposes, its goals, its plans, its progress, and its status. Each of these models interrelates the past, present, and future states of affairs of (1) himself; (2) the group he represents; (3) his boss; (4) the project.

Many of the primary data the participants bring to the meeting are in undigested and uncorrelated form. To each participant, his own collections of data are interesting and important in and of themselves. And they are more than files of facts and recurring reports. They are strongly influenced by insight, subjective feelings, and educated guesses. Thus, each individual's data are reflected in his mental model. Getting his colleagues to incorporate his data into their models is the essence of the communications task.

Suppose you could see the models in the minds of two would-be communicators at this meeting. You could tell, by observing their models, whether or not communication was taking place. If, at the outset, their two models were similar in structure but different simply in the values of certain parameters, then communication would cause convergence toward a common pattern. That is the easiest and most frequent kind of communication.



When mental models are dissimilar, the achievement of communication might be signaled by changes in the structure of one of the models, or both of them.

If the two mental models were structurally dissimilar, then the achievement of communication would be signaled by structural changes in one of the models or in both of them. We might conclude that one of the communicating parties was having insights or trying out new hypotheses in order to begin to understand the other—or that both were restructuring their mental models to achieve commonality.

The meeting of many interacting minds is a more complicated process. Suggestions and recommendations may be elicited from all sides. The interplay may produce, not just a solution to a problem, but a new set of rules for solving problems. That, of course, is the essence of creative interaction. The process of maintaining a current model has within it a set of changing or changeable rules for the processing and disposition of information.

The project meeting we have just described is representative of a broad class of human endeavor which may be described as creative informational activity. Let us differentiate this from another class which we will call informational housekeeping. The latter is what computers today are used for in the main; they process payroll checks, keep track of bank balances, calculate

orbits of space vehicles, control repetitive machine processes, and maintain varieties of debit and credit lists. Mostly they have *not* been used to make coherent pictures of not well understood situations.

We referred earlier to a meeting in which the participants interacted with each other through a computer. That meeting was organized by Doug Engelbart of Stanford Research Institute and was actually a progress-review conference for a specific project. The subject under discussion was rich in detail and broad enough in scope that no one of the attendees, not even the host, could know all the information pertaining to this particular project.

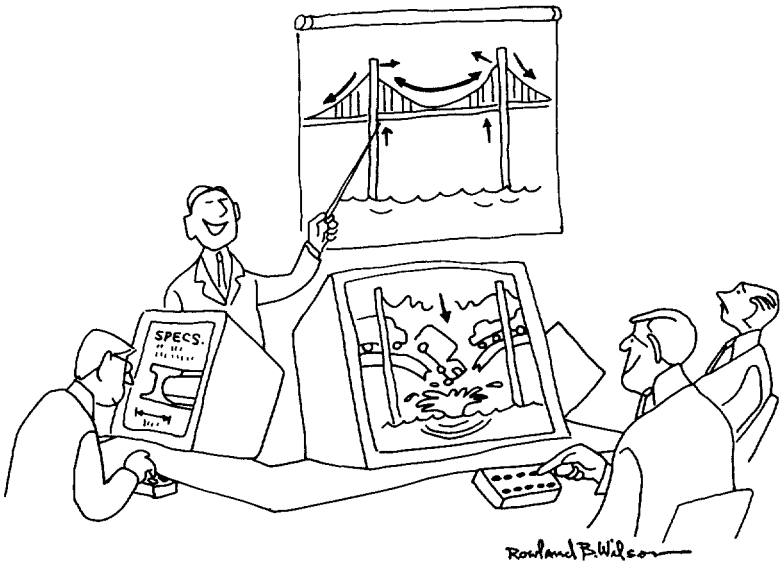
Face to face through a computer

Tables were arranged to form a square work area with five on a side. The center of the area contained six television monitors which displayed the alphanumeric output of a computer located elsewhere in the building but remotely controlled from a keyboard and a set of electronic pointer controllers called "mice." Any participant in the meeting could move a near-by mouse, and thus control the movements of a tracking pointer on the TV screen for all other participants to see.

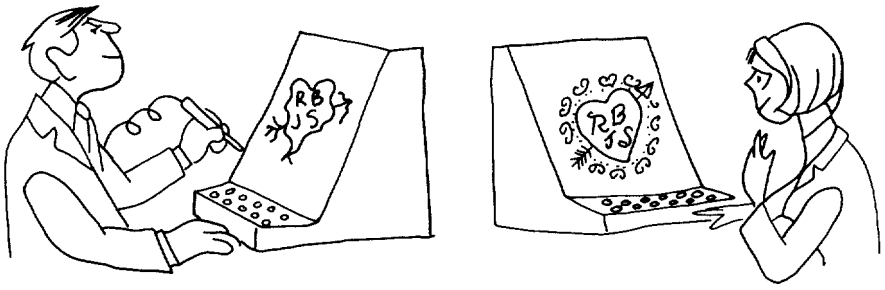
Each person working on the project had prepared a topical outline of his particular presentation for the meeting, and his outline appeared on the screens as he talked—providing a broad view of his own model. Many of the outline statements contained the names of particular reference files which the speaker could recall from the computer to appear in detail on the screens, for, from the beginning of the project, its participants had put their work into the computer system's files.

So the meeting began much like any other meeting in the sense that there was an overall list of agenda and that each speaker had brought with him (figuratively in his briefcase but really within the computer) the material he would be talking about.

The computer system was a significant aid in exploring the depth and breadth of the material. More detailed information could be displayed when facts had to be pinpointed; more global information could be displayed to answer questions of relevance and interrelationship. A future version of this system will make it possible for each participant, on his own TV screen, to thumb through the speaker's files as the speaker talks—and thus check out incidental questions without interrupting the presentation for substantiation.



At a project meeting held through a computer, you can thumb through the speaker's primary data without interrupting him to substantiate or explain.



A communication system should make a positive contribution to the discovery and arousal of interests.

Obviously, collections of primary data can get too large to digest. There comes a time when the complexity of a communications process exceeds the available resources and the capability to cope with it; and at that point one has to simplify and draw conclusions.

It is frightening to realize how early and drastically one does simplify, how prematurely one does conclude, even when the stakes are high and when the transmission facilities and information resources are extraordinary. Deep modeling to communicate—to understand—requires a huge investment. Perhaps even governments cannot afford it yet.

But someday governments may not be able *not* to afford it. For, while we have been talking about the communicant ion process as a cooperative modeling effort in a mutual environment, there is also an aspect of communication with or about an uncooperative opponent. As nearly as we can judge from reports of recent international crises, out of the hundreds of alternatives that confronted the decision makers at each decision point or ply in the "game," on the average only a few, and never more than a few dozen could be considered, and only a few branches of the game could be explored deeper than two or three such plies before action had to be taken. Each side was busy trying to model what the other side might be up to-but modeling takes time, and the pressure of events forces simplification even when it is dangerous.

Whether we attempt to communicate across a division of interests, or whether we engage in a cooperative effort, it is clear that we need to be able to model faster and to greater depth. The importance of improving decision-making processes—not only in government, but throughout business and the professions—is so great as to warrant every effort.

The computer—switch or interactor?

As we see it, group decision-making is simply the active, executive, effect-producing aspect of the kind of communication we are discussing. We have commented that one must oversimplify. We have tried to say why one must oversimplify. But we should not oversimplify the main point of this article. We can say with genuine and strong conviction that a particular form of digital computer organization, with its programs and its data, constitutes the dynamic, moldable medium that can revolutionize the art of modeling and that in so doing can improve the effectiveness of communication among people so much as perhaps to revolutionize that also.

But we must associate with that statement at once the qualification that

the computer alone can make no contribution that will help us, and that the computer with the programs and the data that it has today can do little more than suggest a direction and provide a few germinal examples. Emphatically we do *not* say: "Buy a computer and your communication problems will be solved."

What we do say is that we, together with many colleagues who have had the experience of working on-line and interactively with computers, have already sensed more responsiveness and facilitation and "power" than we had hoped for, considering the inappropriateness of present machines and the primitiveness of their software. Many of us are therefore confident (some of us to the point of religious zeal) that truly significant achievements, which will markedly improve our effectiveness in communication, now are on the horizon.

Many communications engineers, too, are presently excited about the application of digital computers to communication. However, the function they want computers to implement is the switching function. Computers will either switch the communication lines, connecting them together in required configurations, or switch (the technical term is "store and forward") messages.

The switching function is important but it is not the one we have in mind when we say that the computer can revolutionize communication. We are stressing the modeling function, not the switching function. Until now, the communications engineer has not felt it within his province to facilitate the modeling function, to make an interactive, cooperative modeling facility. Information transmission and information processing have always been carried out separately and have become separately institutionalized. There are strong intellectual and social benefits to be realized by the melding of these two technologies. There are also, however, powerful legal and administrative obstacles in the way of any such melding.

Distributed intellectual resources

We have seen the beginnings of communication through a computer—communication among people at consoles located in the same room or on the same university campus or even at distantly separated laboratories of the same research and development organization. This kind of communication—through a single multiaccess computer with the aid of telephone lines—is beginning to foster cooperation and promote coherence more effectively than do present arrangements for sharing computer programs by exchanging

magnetic tapes by messenger or mail. Computer programs are very important because they transcend mere "data''—they include procedures and processes for structuring and manipulating data. These are the main resources we can now concentrate and share with the aid of the tools and techniques of computers and communication, but they are only a part of the whole that we can learn to concentrate and share. The whole includes raw data, digested data, data about the location of data—and documents —and most especially models.

To appreciate the import ante the new computer-aided communication can have, one must consider the dynamics of "critical mass," as it applies to cooperation in creative endeavor. Take any problem worthy of the name, and you find only a few people who can contribute effectively to its solution. Those people must be brought into close intellectual partnership so that their ideas can come into contact with one another. But bring these people together physically in one place to form a team, and you have trouble, for the most creative people are often not the best team players, and there are not enough top positions in a single organization to keep them all happy. Let them go their separate ways, and each creates his own empire, large or small, and devotes more time to the role of emperor than to the role of problem solver. The principals still get together at meetings. They still visit one another. But the time scale of their communication stretches out, and the correlations among mental models degenerate between meetings so that it may take a year to do a week's communicating. There has to be some way of facilitating communicant ion among people wit bout bringing them together in one place.

A single multiaccess computer would fill the bill if expense were no object, but there is no way, with a single computer and individual communication lines to several geographically separated consoles, to avoid paying an unwarrantedly large bill for transmission. Part of the economic difficulty lies in our present communications system. When a computer is used interactively from a typewriter console, the signals transmitted between the console and the computer are intermittent and not very frequent. They do not require continuous access to a telephone channel; a good part of the time they do not even require the full information rate of such a channel. The difficulty is that the common carriers do not provide the kind of service one would like to have---a service that would let one have ad lib access to a channel for short intervals and not be charged when one is not using the channel.

It seems likely that a store-and-forward (i.e., store-for-just-a-moment-

and-forward-right-away) message service would be best for this purpose, whereas the common carriers offer, instead, service that sets up a channel for one's individual use for a period not shorter than one minute.

The problem is further complicated because interaction with a computer via a fast and flexible graphic display, which is for most purposes far superior to interaction through a slow-printing typewriter, requires markedly higher information rates. Not necessarily more information, but the same amount in faster bursts—more difficult to handle efficiently with the conventional common-carrier facilities.

It is perhaps not surprising that there are incompatibilities between the requirements of computer systems and the services supplied by the common carriers, for most of the common-carrier services were developed in support of voice rather than digital communication. Nevertheless, the incompatibilities are frustrating. It appears that the best and quickest way to overcome them—and to move forward the development of interactive *communities* of geographically separated people—is to set up an experimental network of multiaccess computers. Computers would concentrate and interleave the concurrent, intermittent messages of many users and their programs so as to utilize wide-band transmission channels continuously and efficiently, with marked reduction in overall cost.

Computer and information networks

The concept of computers connected to computers is not new. Computer manufacturers have successfully installed and maintained interconnected computers for some years now. But the computers in most instances are from families of machines compatible in both software and hardware, and they are in the same location. More important, the interconnected computers are not interactive, general-purpose, multiaccess machines of the type described by David [1] and Licklider [2]. Although more interactive multiaccess computer systems are being delivered now, and although more groups plan to be using these systems within the next year, there are at present perhaps only as few as half a dozen interactive multiaccess computer *communities.*

These communities are socio-technical pioneers, in several ways out ahead of the rest of the computer world: What makes them so? First, some of their members are computer scientists and engineers who understand the concept of man-computer interaction and the technology of interactive multiaccess systems. Second, others of their members are creative people in other fields

and disciplines who recognize the usefulness and who sense the impact of interactive multiaccess computing upon their work. Third, the communities have large multiaccess computers and have learned to use them. And, fourth, their efforts are regenerative.

In the half-dozen communities, the computer systems research and development and the development of substantive applications mutually support each other. They are producing large and growing resources of programs, data, and know-how. But we have seen only the beginning. There is much more programming and data collect ion—and much more learning how to cooperate-to be done before the full potential of the concept can be realized.

Obviously, multiaccess systems must be developed interactively. The systems being built must remain flexible and open-ended throughout the process of development, which is evolutionary.

Such systems cannot be developed in small ways on small machines. They require large, multiaccess computers, which are necessarily complex. Indeed, the sonic barrier in the development of such systems is complexity.

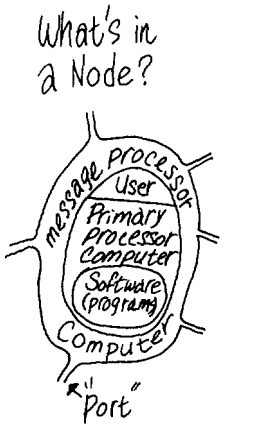
These new computer systems we are describing differ from other computer systems advertised with the same labels: interactive, time-sharing, multiaccess. They differ by having a greater degree of open-endedness, by rendering more services, and above all by providing facilities that foster a working sense of community among their users. The commercially available time-sharing services do not yet offer the power and flexibility of soft ware resources—the "general purposeness''—of the interactive multiaccess systems of the System Development Corporation in Santa Monica, the University of California at Berkeley, Massachusetts Institute of Technology in Cambridge and Lexington, Mass.—which have been collectively serving about a thousand people for several years.

The thousand people include many of the leaders of the ongoing revolution in the computer world. For over a year they have been preparing for the transition to a radically new organization of hardware and software, designed to support many more simultaneous users than the current systems, and to offer them—through new languages, new file-handling systems, and new graphic displays—the fast, smooth interaction required for truly effective man-computer partnership.

Experience has shown the importance of making the response time short and the conversation free and easy. We think those attributes will be almost as important for a network of computers as for a single computer.

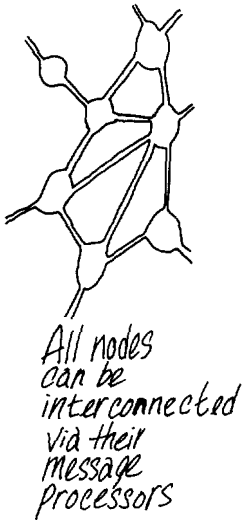
Today the on-line communities are separated from one another function-

ally as well as geographically. Each member can look only to the processing, storage and software capability of the facility upon which his community is centered. But now the move is on to interconnect the separate communities and thereby transform them into, let us call it, a supercommunity. The hope is that interconnection will make available to all the members of all the communities the programs and data resources of the entire supercommunity. First, let us indicate how these communities can be interconnected; then we shall describe one hypothetical person's interaction with this network, of interconnected computers.



Message processing

The hardware of a multiaccess computer system includes one or more central processors, several kinds of memory—core, disks, drums, and tapes—and many consoles for the simultaneous on-line users. Different users can work simultaneously on diverse tasks. The software of such a system includes supervisory programs (which control the whole operation), system programs for interpretation of the user's commands, the handling of his files, and graphical or alphanumeric display of information to him (which permit people not skilled in the machine's language to use the system effectively), and programs and data created by the users themselves. The collection of people, hardware, and software-the multiaccess computer together with its local community of users—will become a node in a geographically distributed computer network. Let us assume for a moment that such a network has been formed.



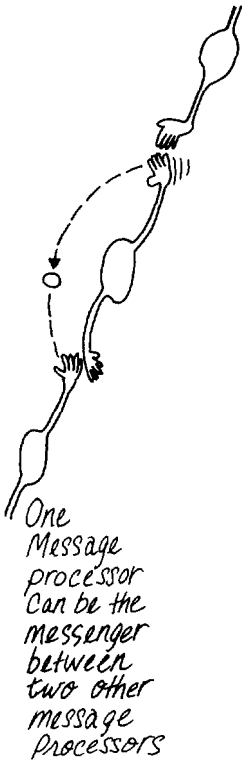
For each node there is a small, general-purpose computer which we shall call a "message processor." The message processors of all the nodes are interconnected to form a fast store-and-forward network. The large multiaccess computer at each node is connected directly to the message processor there. Through the network of message processors, therefore, all the large computers can communicate with one another. And through them, all the members of the supercommunity can communicate-with other people, with programs, with data, or with selected combinations of those resources. The message processors, being all alike, introduce an element of uniformity into an otherwise grossly nonuniform situation, for they facilitate both hardware and software compatibility among diverse and poorly compatible computers. The links among the message processors are transmission and high-speed *digital* switching facilities provided by common carrier. This allows the linking of the message processors to be reconfigured in response to demand.

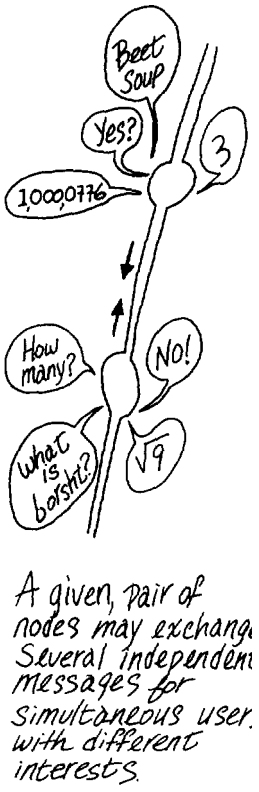
A message can be thought of as a short sequence of "bits" flowing through the network from one multiaccess computer to another. It consists of two types of information: control and data. Control information guides the transmission of data from source to destination. In present transmission systems, errors are too frequent for many computer applications. However, through the use of error detection and correction or retransmission procedures in the message processors, messages can be delivered to their destinations intact even though many of their "bits" were mutilated at one point or another along the way. In short, the message processors function in the system as traffic directors, controllers, and correctors.

Today, programs created at one installation on a given manufacturer's computer are generally not of much value to users of a different manufacturer's computer at another installation. After learning (with difficulty) of a distant program's existence, one has to get it, understand it, and recode it for his own computer. The cost is comparable to the cost of preparing a new program from scratch, which is, in fact, what most programmers usually do. On a national scale, the annual cost is enormous. Within a network of interactive, multiaccess computer systems, on the other hand, a person at one node will have access to programs running at other nodes, even though those programs were written in different languages for different computers.

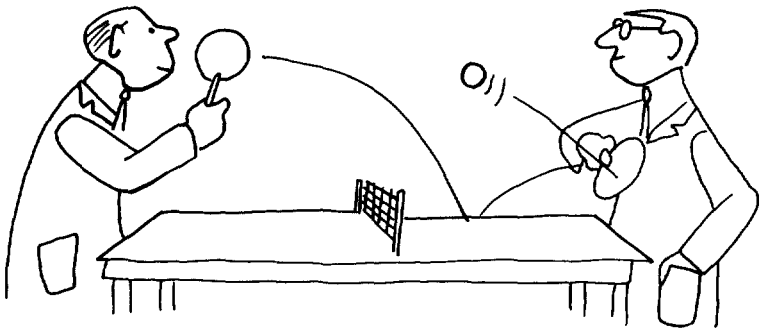
The feasibility of using programs at remote locations has been shown by the successful linking of the AN/FSQ-32 computer at Systems Development Corporation in Santa Monica, Calif., with the TX-2 computer across the continent at the Lincoln Laboratory in Lexington, Mass. A person at a TX-2 graphic console can make use of a unique list-processing program at SDC, which would be prohibitively expensive to translate for use on the TX-2. A network of 14 such diverse computers, all of which will be capable of sharing one another's resources, is now being planned by the Defense Department's Advanced Research Projects Agency, and its contractors.

The system's way of managing data is crucial to the user who works in interaction with many other people. It should put generally useful data, if not subject to control of access, into public files. Each user, however, should have complete control over his personal files. He should define and distribute the "keys" to each such file, exercising his option to exclude all others from any kind of access to it; or to permit anyone to "read" but not modify or execute it; or to permit selected individuals or groups to execute but not read it; and so on—with as much detailed specification or as much aggregation as he likes. The system should provide for group and organizational files within its overall information base.





A given pair of nodes may exchange several independent messages for simultaneous users with different interests.



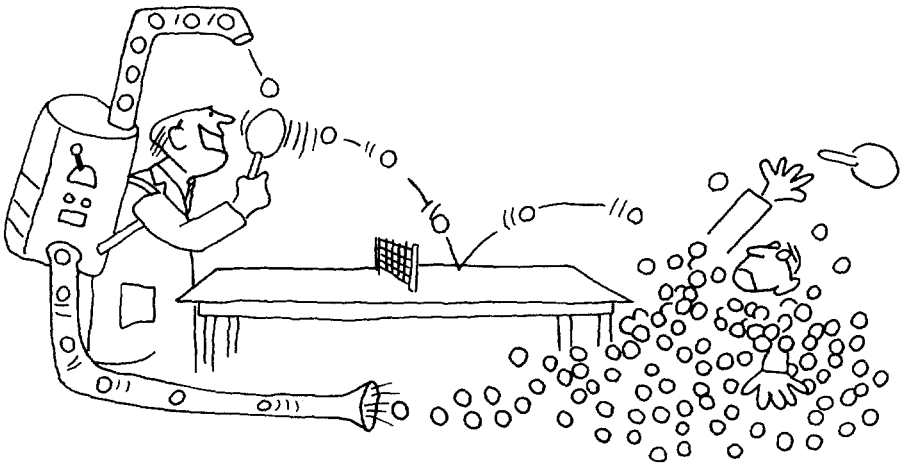
Interactive communication consists of short spurts of dialog . . . . .

At least one of the new multiaccess systems will exhibit such features. In several of the research centers we have mentioned, security and privacy of information are subjects of active concern; they are beginning to get the attention they deserve.

In a multiaccess system, the number of consoles permitted to use the computer simultaneously depends upon the load placed on the computer by the users' jobs, and may be varied automatically as the load changes. Large general-purpose multiaccess systems operating today can typically support 20 to 30 simultaneous users. Some of these users may work with low-level "assembly" languages while others use higher-level "compiler" or "interpreter" languages. Concurrently, others may use data management and graphical systems. And so on.

But back to our hypothetical user. He seats himself at his console, which may be a terminal keyboard plus a relatively slow printer, a sophisticated graphical console, or any one of several intermediate devices. He dials his local computer and "logs in" by presenting his name, problem number, and password to the monitor program. He calls for either a public program, one of his own programs, or a colleague's program that he has permission to use. The monitor links him to it, and he then communicates with that program.

When the user (or the program) needs service from a program at another



. . . filibustering destroys communication.

node in the network, he (or it) requests the service by specifying the location of the appropriate computer and the identity of the program required. If necessary, he uses computerized directories to determine those data. The request is translated by one or more of the message processors into the precise language required by the remote computer's monitor. Now the user (or his local program) and the remote program can interchange information. When the information transfer is complete, the user (or his local program) dismisses the remote computer, again with the aid of the message processors. In a commercial system, the remote processor would at this point record cost information for use in billing.

Who can afford it?

The mention of billing brings up an important matter. Computers and long-distance calls have "expensive" images. One of the standard reactions to the idea of "on-line communities" is: "It sounds great, but who can afford it ?"

In considering that question, let us do a little arithmetic. The main elements of the cost of computer-facilitated communication, over and above the salaries of the communicators, are the cost of the consoles, processing, storage, transmission, and supporting software. In each category, there is a

wide range of possible costs, depending in part upon the sophistication of the equipment, programs, or services employed and in part upon whether they are custom-made or mass-produced.

Making rough estimates of the hourly component costs per user, we arrived at the following: $1 for a console, $5 for one man's share of the services of a processor, 70 cents for storage, $3 for transmission via line leased from a common carrier, and $1 for software support—a total cost of just less than $11 per communicator hour.

The only obviously untenable assumption underlying that result, we believe, is the assumption that one's console and the personal files would be used 160 hours per month. All the other items are assumed to be shared with others, and experience indicates that time-sharing leads on the average to somewhat greater utilization than the 160 hours per month that we assumed, Note, however, that the console and the personal files are items used also in individual problem solving and decision making. Surely those activities, taken together with communication, would occupy at least 25% of the working hours of the on-line executive, scientist or engineer. If we cut the duty factor of the console and files to one quarter of 160 hours per month, the estimated total cost comes to $16 per hour.

Let us assume that our $16/hr interactive computer link is set up between Boston, Mass., and Washington, D.C. Is $16/hr affordable? Compare it first with the cost of ordinary telephone communication: Even if you take advantage of the lower charge per minute for long calls, it is less than the daytime direct-dial station-to-station toll. Compare it with the cost of travel: If one flies from Boston to Washington in the morning and back in the evening, he can have eight working hours in the capital city in return for about $64 in air and taxi fares plus the spending of four of his early morning and evening hours en route. If those four hours are worth $16 each, then the bill for the eight hours in Washington is $128—again $16 per hour. Or look at it still another way: If computer-aided communication doubled the effectiveness of a man paid $16 per hour then, according to our estimate, it would be worth what it cost if it could be bought right now. Thus we have some basis for arguing that computer-aided communication is economically feasible. But we must admit that the figure of $16 per hour sounds high, and we do not want to let our discussion depend upon it.

Fortunately, we do not have to, for the system we envision cannot be bought at this moment. The time scale provides a basis for genuine optimism about the cost picture. It will take two years, at least, to bring the first interactive computer networks up to a significant level of experimental

activity. Operational systems might reach critical size in as little as six years if everyone got onto the bandwagon, but there is little point in making cost estimates for a nearer date. So let us take six years as the target.

In the computer field, the cost of a unit of processing and the cost of a unit of storage have been dropping for two decades at the rate of 50% or more every two years. In six years, there is time for at least three such drops, which cut a dollar down to 12 1/2 cents. Three halvings would take the cost of processing, now $5 per hour on our assumptions, down to less than 65 cents per hour.

Such advances in capability, accompanied by reduction in cost, lead us to expect that computer facilitation will be affordable before many people are ready to take advantage of it. The only areas that cause us concern are consoles and transmission.

In the console field, there is plenty of competition; many firms have entered the console sweepstakes, and more are entering every month. Lack of competition is not the problem. The problem is the problem of the chicken and the egg—in the factory and in the market. If a few companies would take the plunge into mass manufacture, then the cost of a satisfactory console would drop enough to open up a mass market. If large on-line communities were already in being, their mass market would attract mass manufacture. But at present there is neither mass manufacture nor a mass market, and consequently there is no low-cost console suitable for interactive on-line communication.

In the field of transmission, the difficulty may be lack of competition. At any rate, the cost of transmission is not falling nearly as fast as the cost of processing and storage. Nor is it falling nearly as fast as we think it should fall. Even the advent of satellites has affected the cost picture by less than a factor of two. That fact does not cause immediate distress because (unless the distance is very great) transmission cost is not now the dominant cost. But, at the rate things are going, in six years it will be the dominant cost. That prospect concerns us greatly and is the strongest damper to our hopes for near-term realization of operationally significant interactive networks and significant on-line communities.

On-line interactive communities

But let us be optimistic. What will on-line interactive communities be like? In most fields they will consist of geographically separated members, sometimes grouped in small clusters and sometimes working individually. They

will be communities not of common location, but of *common interest.* In each field, the overall community of interest will be large enough to support a comprehensive system of field-oriented programs and data.

In each geographical sector, the total number of users—summed over all the fields of interest—will be large enough to support extensive general-purpose information processing and storage facilities. All of these will be interconnected by telecommunications channels. The whole will constitute a labile network of networks—ever-changing in both content and configuration.

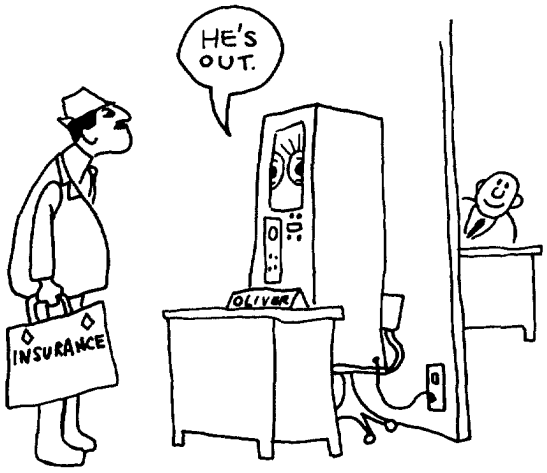
What will go on inside? Eventually, every informational transaction of sufficient consequence to warrant the cost. Each secretary's typewriter, each data-gathering instrument, conceivably each dictation microphone, will feed into the network.

You will not send a letter or a telegram; you will simply identify the people whose files should be linked to yours and the parts to which they should be linked-and perhaps specify a coefficient of urgency. You will seldom make a telephone call; you will ask the network to link your consoles together,

You will seldom make a purely business trip, because linking consoles will be so much more efficient. When you do visit another person with the object of intellectual communication, you and he will sit at a two-place console and interact as much through it as face to face. If our extrapolation from Doug Engelbart's meeting proves correct, you will spend much more time in computer-facilitated teleconferences and much less en route to meetings.

A very important part of each man's interaction with his on-line community will be mediated by his OLIVER. The acronym OLIVER honors Oliver Selfridge, originator of the concept. An OLIVER is, or will be when there is one, an "on-line interactive vicarious expediter and responder," a complex of computer programs and data that resides within the network and acts on behalf of its principal, taking care of many minor matters that do not require his personal attention and buffering him from the demanding world. "You are describing a secretary," you will say. But no! Secretaries will have OLIVERS.

At your command, your OLIVER will take notes (or refrain from taking notes) on what you do, what you read, what you buy and where you buy it. It will know who your friends are, your mere acquaintances. It will know your value structure, who is prestigious in your eyes, for whom you will do what



Your computer will know who is prestigious in your eyes and buffer you from a demanding world.

with what priority, and who can have access to which of your personal files. It will know your organization's rules pertaining to proprietary information and the government's rules relating to security classification.

Some parts of your OLIVER program will be common with parts of other people's OLIVERS; other parts will be custom-made for you, or by you, or will have developed idiosyncrasies through "learning" based on its experience in your service.

Available within the network will be functions and services to which you subscribe on a regular basis and others that you call for when you need them. In the former group will be investment guidance, tax counseling, selective dissemination of information in your field of specialization, announcement of cultural, sport, and entertainment events that fit your interests, etc. In the latter group will be dictionaries, encyclopedias, indexes, catalogues, editing programs, teaching programs, testing programs, programming systems, data bases, and—most important—communication, display, and modeling programs.

All these will be—at some late date in the history of networking—systematized and coherent; you will be able to get along in one basic language up to the point at which you choose a specialized language for its power or terseness.

When people do their informational work "at the console" and "through the network," telecommunication will be as natural an extension of individual work as face-to-face communication is now. The impact of that fact, and of the marked facilitation of the communicative process, will be very great—both on the individual and on society.

First, life will be happier for the on-line individual because the people with whom one interacts most strongly will be selected more by commonality of interests and goals than by accidents of proximity. Second, communication will be more effective and productive, and therefore more enjoyable. Third, much communication and interaction will be with programs and programmed models, which will be (a) highly responsive, (b) supplementary to one's own capabilities, rather than competitive, and (c) capable of representing progressively more complex ideas without necessarily displaying all the levels of their structure at the same time-and which will therefore be both challenging and rewarding. And, fourth, there will be plenty of opportunity for everyone (who can afford a console) to find his calling, for the whole world of information, with all its fields and disciplines, will be open to him—with programs ready to guide him or to help him explore.

For the society, the impact will be good or bad, depending mainly on the question: Will "to be on line" be a privilege or a right? If only a favored segment of the population gets a chance to enjoy the advantage of "intelligence amplification," the network may exaggerate the discontinuity in the spectrum of intellectual opportunity.

On the other hand, if the network idea should prove to do for education what a few have envisioned in hope, if not in concrete detailed plan, and if all minds should prove to be responsive, surely the boon to humankind would be beyond measure.

Unemployment would disappear from the face of the earth forever, for consider the magnitude of the task of adapting the network's software to all the new generations of computer, coming closer and closer upon the heels of their predecessors until the entire population of the world is caught up in an infinite crescendo of on-line interactive debugging.

Acknowledgements

Evan Herbert edited the article and acted as intermediary during its writing between Licklider in Boston and Taylor in Washington.
Roland B. Wilson drew the cartoons to accompany the original article.

References

[1] Edward E. David, Jr., "Sharing a Computer," *International Science and Technology,* June, 1966.

[2] J. C. R. Licklider, "Man-Computer Partnership," *International Science and Technology,* May, 1965.