

THE UNIVERSITY OF MICHIGAN
INDUSTRY PROGRAM OF THE COLLEGE OF ENGINEERING

**THE IMPACT OF REMOTE-TERMINAL
COMPUTERS ON ENGINEERING
DEPARTMENTS IN INDUSTRY**

*Transcript of the Eleventh Ann Arbor
Industry-Education Symposium*



June, 1967
IP-781

TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	
R. E. Carroll Director, Industry Program	1
WELCOME	
Gordon J. Van Wylen Dean, College of Engineering.....	3
INTRODUCTION TO THE CONCEPT OF TIMESHARING	
Norman R. Scott Associate Dean, College of Engineering	5
TIMESHARING AT THE FORD TECHNICAL COMPUTING CENTER	
Charles W. Missler Ford Motor Company	17
THE COMPUTER UTILITY - ON-LINE COMPUTER SERVICE AT RETAIL	
Charles W. Adams C. W. Adams Associates, Inc. and Keydata Corp.....	43
AFTERNOON SESSION	
EMANCIPATION OF ENGINEERING FROM FINANCE-ORIENTED COMPUTERS	
Richard F. Lewis Clark Equipment Company	63
REMOTE TERMINAL EQUIPMENT FOR TOMORROW'S DESIGN ENGINEER	
Bertram Herzog The University of Michigan	77

INTRODUCTION

R. E. Carroll

Director, Industry Program

Just four and a half years ago the Industry Program sponsored a symposium on the Present and Possible Future Roles of Computers. It was a very fine meeting, but it is most interesting to note, in reviewing the symposium proceedings, that the terms timesharing, remote terminals, on-line, and real-time were not mentioned. The exploding technology represented by such terms will have a tremendous impact on almost every phase of engineering. We hope that your experience here today will be both interesting and valuable.

WELCOME

Gordon J. Van Wylene

Dean, College of Engineering
The University of Michigan

Thank you, Ray. It is both my pleasure and privilege to extend the welcome of the College of Engineering to each of you attending this Industry Program Symposium today. I believe that the first occasion I had to perform this function as the Dean of the College of Engineering was at a similar meeting one year ago. I was new at that time and was attempting to learn what was involved in this job. This past year has been a very interesting and important experience for me, as I have tried to learn about the college as a whole.

One of the things which we have discussed at considerable length this past year is the relationship between the College of Engineering and Industry. We recognize that technology is moving at a tremendous pace and it is imperative that we keep in close contact with industry, and to be fully aware of the forefronts of technology. We are learning from you and at the same time we are attempting to provide a current and relevant education to the students. We want to make sure that our research programs are related to what is going in industry and to contribute directly to them. In other words, what we need at this time is a close partnership with industry interacting on both teaching and research.

This is particularly related to the whole matter of Continuing Engineering Education which one reads a great deal about these days. I am happy to report that the building which you see rising just to your right is the Chrysler Center for Continuing Engineering Education. This will be completed next August, if the schedule holds, and we are looking forward to this event. Should you attend the next annual meeting of the Industry Program we anticipate having it in this fine building. Perhaps it will be the first meeting we will hold there next fall. This program of Continuing Engineering Education is, at present, referred to as the Engineering Summer Conferences. Last summer we had 40 such conferences; most of them were two week with a few of them one week sessions. We had over 2600 persons attending these 40 conferences. We are looking forward, with the completion of this new building, to be able to present these conferences throughout the year as well as during the summer. We are limited to the summer at the present time because we do not have the facilities to have these meetings simultaneously with our regular academic program. In the College of Engineering we have more than 3200

undergraduates and 1150 graduate students attending this fall. This utilizes every bit of our facilities so we are forced, therefore, to schedule a meeting such as this in a room which is not designed as a classroom or a teaching room per se.

Again, may I tell you how glad we are that you are here today. We hope that you find this program enjoyable and profitable and look forward to continuing interaction with you in a variety of ways. I should like to express my appreciation to Ray Carroll for all the work which he has done on the Industry Program in making this program possible. It is now my privilege to introduce the Associate Dean of the College of Engineering, Norman Scott. Computer technology and computers is his field so he not only is chairing this session in the capacity of Associate Dean but more particularly on the basis of his technical qualifications. We tend to supplement each other in many ways. He was born in the East and did his undergraduate work at MIT then went to Illinois for his doctorate. I was born in the Midwest, did my undergraduate work here and then went to MIT for graduate study. Our paths have again crossed here at Michigan and we have enjoyed working together this past year. If you have not become acquainted with Norm Scott yet, I hope you will take advantage of the opportunity today. So, Norman, I will turn the meeting over to you at this time.

INTRODUCTION TO THE CONCEPT OF TIMESHARING

Norman R. Scott

Associate Dean, College of Engineering
The University of Michigan
Chairman of the Symposium

INTRODUCTION TO THE CONCEPT OF TIMESHARING

Thank you, Dean Van Wylen.

Let me add a few words of my own to those Dean Van Wylen has already extended to you by way of welcome. We are very happy¹ to have so many of you with us today. I hope you find this a pleasurable and worthwhile occasion. Those of us who are in the computer field get a great delight out of talking about computers, and I hope you will bear with us as we, at least, enjoy the program. I hope you, too, will find some benefits in it.

My role today is to paint for you a broad picture of what is meant by "Timesharing." I am going to give you the broad-brush background features of the picture. Some of the more detailed, prominent foreground aspects of this picture will be painted for you by the various other speakers who will describe some actual experiences to help explain what they mean by "Timesharing" systems.

One of the things that you should become acquainted with early in the timesharing business is the jargon that is used in this field. It is too bad there is a jargon, but there is! I think every field has its own special idiom. You will hear words such as "Timesharing," "On-line," "Real Time," "Computer Utility," and "The Conversational Mode." In fact, as I look at the program for today, I see the term "timesharing" appears twice; we talk twice about "remote-terminal," and once about "computer utility." These are some of the words we will gradually expand upon and discuss with you today. What they have as their common meaning is a close user-interaction with the machine, as though the men who were using the machine were wholly in charge of it; as though the machine belonged solely to them. This is one of the principal things that timesharing implies, but it also implies a lot of other things. We will gradually fill these in for you as the day goes on.

First, however, let me give you a little of the history of how the computing business got to the state it is in today, or at least how the art of using computers got to be the particular style of technology that it is today.

The early computers, those that existed in the period from 1948 to 1952 or so, were mostly one-of-a-kind computers. Many of them were experimental machines that tended to be used by people who knew them intimately and who could interact closely with them in solving their own problems. One such machine was the MIDAC, which we had at our Willow Run Laboratories for several years. Another was the Whirlwind computer which Charlie Adams, one of our speakers, knew intimately back at that

period. Several other machines which existed in various universities around the country were of this one-of-a-kind class. The people who worked with these machines knew them in such intimate detail that when some strange result occurred at the console, the user could quickly find out whether it was the computer that was at fault or his own program. He could sit at the console, operate the machine, change the course of the computation, and interact very closely with the computation as it went forward. In many ways, this was an ideal way to operate with the machine.

It is delightful to have a powerful tool like this available wholly to you as the user. However, the gradual trend was away from this and moved in the direction of optimizing the efficiency of the machine. To have the user sit at the machine and make sole use of it is fine for that user, but people came to realize that while one man was sitting thinking about his problem another man could not be using the machine. So, with the development of somewhat more powerful languages and somewhat more powerful systems for making use of the machine, we gradually moved the user a little farther away and put the operation of it in charge of a skilled technician, who knew what buttons to press, and how to put programs in and out; but we isolated the user from the machine. The user now became a person who would prepare a program, have it punched out, and then transmit the cards to the computing center. The computing center, in turn, would run the problem for him on its own schedule. The reasons for doing this were fairly obvious. Computers were expensive, and if one wanted to optimize the efficiency of a machine that cost several hundred dollars an hour to use, one could not very well afford to have people sitting there wasting its time. So, from the standpoint of computational efficiency and cost considerations we moved the user away from the machine.

As you see, we have now swung full course, and we have come around to the other consideration that what we want to optimize is the time of the user rather than the time of the machine. But this was not possible in the early days of the use of the computer. People gradually evolved languages which made it possible for the user to describe his problem in simpler terms and operating systems which made it possible for the computer itself to shuffle user problems in and out and execute them in turn and perform the various printouts that were required. We thus came around to the mode of using computers in what is now called batch-mode processing. A typical way in which one does this is as follows: the user programs his problem, keypunches his card deck, and puts the deck in a collection box. Some place in the system, during the day, a messenger comes around, picks up the cards and carries them to the computing center. The operator at the computing center then uses a subsidiary computer to read these cards into the machine, taking

several decks from several users and putting them, in sequence, on magnetic tape. At his own convenience, the operator later uses this magnetic tape, run it, prepares a new magnetic tape with results, then transmit this to the subsidiary computer which prepares the output format. In this way we have totally isolated the user from the actual computer. In fact, we have even relieved the main computer of the burden of doing the input and output editing of all the details of reading and printing information in various formats and styles.

This does, indeed, optimize the use of the computer, but it has the disadvantage that the turn-around time becomes a large and significant factor. In fact, here at the University, we found that turn-around time - the time from the preparation of the original punched card deck to receipt of the completed information - was usually of the order of one day. Fortunate users could, once in a while, get a problem through and back, put in a revision and get the answer to that back and manage two complete runs in one day. At saturation peak seasons, it often was difficult to get one run a day. People sometimes found that it took two days to get a problem back.

Note that from the user's point of view, the messenger service is part of the computer in that arrangement. The user, having prepared a card deck, transmits it to this large entity which is the computer. Within this large entity there is a messenger who runs from where the cards were back to the computing center and then brings the result to the box where the user picks them up. To have this kind of human messenger service in the computing system is indeed a rather awkward arrangement. This is one of the factors which cause the turn-around time to be so severe.

One of the major impetuses in establishing this particular mode of operation was the fact that computing power became fairly widespread, in the middle 1950's, yet not so widespread that everyone could have a computer at his finger tips. Much of the early operating experience on batch-mode processors was accumulated on the IBM 650 machine, of which several thousand copies were made. This machine was in use in many laboratories, university establishments, and business organizations throughout the mid 1950 period. The trends at that time were in several directions. I think I can attribute the current state of the art to several main developments which took place. First, we had the vastly increased speed of computers and the reduced cost of computation. This is not to say that computers necessarily cost less per computer. In fact, the bigger the computer, the more the cost of the machine per hour of computation. But at the same time, the bigger the computer, the greater the number of computations performed per unit of time. Although, for a very large machine the cost per hour of computation

tends to go up, in fact, the cost per computation was going down steadily all through the 1950's, and this is the trend that has continued even up to today. We see no end of it in sight. Second, the development of much simpler languages for the use of the user has had a significant effect upon the trend of the direction of computation. The early user was the man who knew a lot about the computer, perhaps even helped to design the computer, and could communicate with the machine directly in its own language. Its own language is usually a binary language - a succession of ones and zeros. The early user was frequently very skilled in communicating with the machine in this binary kind of language and wrote his programs in a form which was amenable to a simple translation into binary formats.

This meant that the machine was really restricted to the high priests of the technology, who understood the language and the means of communicating with the computer. In order to broaden the use of computers, people soon recognized that it would be desirable to have a simpler language format. The simplest language of all, of course, would enable one to walk up to the machine and say "Compute my payroll; here it is." Machines can not yet accept that kind of instruction. We have, at the least, to type out some instructions. In the present day we can not even type them out in clear English. We have to restrict our English a bit and dress up the language that we are using with a few restricted format symbols that the machine can recognize. Now the trend is toward simpler and simpler languages which are more nearly like the language that the user himself will employ in describing the problem.

There has been a strong drift toward what are called problem-oriented languages. These are languages which are convenient for the description of particular classes of problems and which the computer in turn can readily translate into the binary information that it needs for its own internal operation. Another factor that has had a significant effect is that the reliability of computers is now enormously greater than it was several years ago. The measure that we often use in describing the reliability of the computer is the mean-free-time between errors; the average time that the computer will run between making a mistake on its own. This is not counting the errors that the user makes in preparing his program. The machine carries out the instruction perfectly but the instructions may be wrong. What I am speaking of as mean-free-time between errors is mean-free-time between machine malfunctions. It is not so many years since the average time between machine malfunctions tended to be about five or ten minutes. This meant that one could get perhaps five minutes of reliable computation before suspecting that something might go wrong pretty soon. Therefore people had to include in their programs all sorts of means for going back and

catching the program at a certain point, such as the last reliable point to which they were sure a computation had run correctly, and then going on from this point. Now mean-free-time between errors runs generally a good many hours. We find that the reliability of the overall system is enormously greater than it was before. You see the same thing in the automotive industry where they are now giving five year and 50,000 mile warranties. An equivalent sort of thing is happening in the computer industry.

Another development is the fact that the memory capacity in the computer is enormously greater now than it was 15 years ago. It is interesting to reflect that our first computer here at the University of Michigan had a total memory capacity of 512 words of storage. Five hundred and twelve separate data groups could be stored away in the memory and retrieved rapidly from that memory. John Von Neumann, the great mathematician and one of the pioneers of the computer business, is said to have remarked about 1950, that he could not really see the need for a computer to have more than about 1,000 words of memory. This was quite a lot larger than most people had at that time. Now the smallest memory that is commercially available is about 4,000 words. You can buy increments of 4,000 words of memory very cheaply. Many computers exist on the market with a standard memory complement of 32,000 words of rapid-access memory, supplemented by many millions of words of somewhat slower access memory. One can now reliably employ very, very large memories indeed.

Along with this there goes an advancing development in display technology. As we want to communicate more and more readily with the computer we find that the printed page is not necessarily the best way to do this communication. The use of graphical input and output consoles, the use of cathode-ray tubes, the use of devices on which we can draw pictures which are then connected and transmitted into the computer where they are manipulated, the use of devices on which the computer can in turn draw pictures for us, have made a substantial difference in the way that the computer is used.

I think that all of these things together have contributed to what we now call timesharing, the concept that many users can share the central computer. The computer now calculates so rapidly that if the user stops to scratch his head for a moment, in that brief time, the computer could have done 100,000 computations for someone else.

One of the concepts of timesharing has been the idea that we will slice up the time of the computer so that every user gets quick reaction as though he were the only person making use of the machine. In effect, we are then interleaving each user's programs among all the other users' programs, some of which may or may not be active at any given moment. On a heavily loaded system where every user is demanding

access simultaneously, the computer can only take the users in turn and thus the important question is, "What is the reaction time?" How soon after you type something into the computer should the computer respond to indicate to you that it has received the message that you have typed in? In the old days, this time ran several hours. We would transmit a deck of cards, the computer would calculate on the deck of cards, and we would get a new deck, or a new print-out back, and this time ran into a good many hours, sometimes a day or two. Nowadays we would like to be able to turn to such a console as this one on the platform, type some information into the computer, then receive, within a matter of a few seconds at the most, some response to indicate that the computer had accepted our transmission and is preparing the work on the information that we had put in. The question of what the reaction time should be in an optimum system is one that has been the subject of a good deal of discussion among computer designers and builders. Some people feel that there ought to be, within a half a second, a response. Some people feel that two or three seconds is adequate. Indeed, it may depend a good deal upon what the particular application is.

This is what we mean by working on-line with the computer. "On-line" is a somewhat vaguely defined term to imply that I have the computer for myself and I am in communication with it on a more or less satisfactory kind of instantaneous give-and-take basis. If I can sit with the computer at my side, ask it questions, and get back answers as though you and I were talking, then I would consider this to be an on-line mode of operation. "On-line" implies considerably more than that. It may imply that there is a process - a manufacturing process - going on in industry, which is connected to a computer that is carrying out computations that can affect the execution of that process. This is a tight kind of coupling of a computer in an on-line sense to a process. The coupling of the computer to a user who types information in and out is not quite so tight. But we think of it also as being on-line, and the user has direct access to the computer.

The conversational aspect of the use of the computer is another one that I think is an important feature of many of the timesharing modes. When we type something into the computer - when we give it a piece of information - we would like the computer to make a response to us that indicates some kind of intelligence at work behind the keyboard. Many computers are indeed programmed to type back fairly intelligent remarks to us in response to the information that we type in. If one works intimately with a computer in this fashion, pretty soon he finds that the typical user regards the computer not as an IT but as a HE. People will speak of the computer as, "He said this to me in response to what I typed in." This is a very interesting kind of reaction. It suggests that there is a good deal of intelligence behind the computer.

There is! It is all intelligence that the people who designed the system to accept the information put in to make it come out with a fairly reasonable set of comments and response to the information. Often the information that comes back from the computer is diagnostic. It indicates the kinds of things that may have gone wrong with the program. You failed to type in some particular set of symbols which the computer requires as information describing the program that you are preparing for. The computer will recognize this situation and type back a canned message to say what symbol you left out.

The main impetus in developing these timesharing systems has been a recognition that perhaps it is not the time of the computer that we ought to optimize so much as the time of the user. Now that computing power has become a good deal cheaper than it was some years ago and we do not have to be quite so concerned about making the computer work at optimum efficiency, we have come around to the point of view that it is the user whose time we want to optimize.

If the computer is to be of service to users, we ought to be able to offer these users some optimum service to minimize the time that they have spent in preparing programs and problems for the machine. This is one of the major things that timesharing buys for us. It buys convenience for the user, rapid response, ability to check out his programs quickly, ability to carry out simple computations, a degree of intimacy we lost when we moved away from the era of about 15 years ago into the batch processing mode. It returns us to a feeling that the computer is available on call as a servant in the carrying out of computations.

In fact, one can often view the computer as a utility. It is something that is standing there waiting to be used when you need it just as the electric utilities in the building are there when you need them at a flip of a switch. If we do not need them, somebody else has access to the system. And like a utility, we would hope that the number of users on the system will never saturate it and cause all the lights to go out or cause the computing power to diminish to zero, although this is certainly a very real possibility.

Another aspect of the use of computers in the timesharing mode is that it is now possible for us to have remote-terminals away from the computer. We need not provide every user with a computer. We need to provide each user only a facility to get access to the central computer, just as in the electrical utility business where we do not provide every user an electrical generator but have a central generator back at a remote point into which each user can get access as he wishes. This teletype here is a good example of this kind of situation. The teletype is not a computer; it is simply a communication device which can have access to any one of quite a wide variety of computers. Whatever one we want, we get access to provided we have already made arrangement to rent some time on that machine.

The sharing of the computer is an important aspect but I should not minimize the fact that we can share common programs and share data banks that may be available in the central computer. We may, indeed, be able to share specialized services which are available in the centralized computer. There exist, at the present day, as many as several dozen of these timesharing computer systems around the country. Some of them are totally internal to large corporations that have scattered remote terminals around their plants and offices and use their own central computer. Some of them are services that are available on hire. One can rent a teletype and rent time on a central computer system. Some of these central computer systems provide very specialized services. In fact, Mr. Adams, who is going to be our second speaker this morning, will be describing to you a particular computer utility which he manages and which provides, primarily, business services. The existence of this variety of patterns of use of the computer gives, I think, a new scope and power to the computer facilities that are available to the user.

The key question that I think all of you gentlemen will want to be asking about this kind of system, is the very hard-headed question, "What is in it for me? What can I do with this kind of system that I could not do before? How can I do the things that I used to do in a better and more expeditious fashion?" Let me point out that I think the most important question to ask here is not merely, "How can I do the things that I have been doing, better than I have been doing them before?" but, "How can I now do with the computer things that I never did before?" The important thing about the computer is not that it speeds up the old ways of doing things but that it allows you to get a "handle" on new techniques for doing things which you never attacked before. Our speakers through the day will give us some insights into these various considerations, and I hope you will come away from this with a feeling for the rich variety of techniques that are available for you in using computers and some of the potentialities that this new mode of computer usage has in your business and your concern. Let me at this point, turn the program over to our speakers.

The first speaker for today will be Mr. Charles W. Missler from the Ford Motor Company. Mr. Missler is a man with a rich background in this field and I think he will present a very interesting picture of operations at the Ford Motor Company. He is a graduate of the U.S. Naval Academy. He spent several years in the aerospace and computer industries, then joined the Ford Motor Company in 1963. In 1964 he was put in charge of the technical review department of the Ford Product Engineering Office and was responsible for the technical auditing of both their advanced and current product program. His present position is Manager of the Technical Computer Center, which provides on-line computer services to Ford operations both in the United States and overseas.

This is a most remarkable system, and has been cited in a recent survey in the Automotive Industry's journal as being the most ambitious and advanced computer center to be found at the present time. Mr. Missler is, among other things, the recipient, in 1964, of the Engineering Society of Detroit's Outstanding Engineer of the Year award. I think you will find him to be a very interesting and well informed speaker - Mr. Missler.

TIMESHARING AT THE FORD TECHNICAL COMPUTING CENTER

Charles W. Missler

Manager

Technical Computer Services Engineering Staff
Ford Motor Company

TIMESHARING AT THE FORD TECHNICAL COMPUTING CENTER

Good-morning. I would like to express my appreciation for having the opportunity to represent the Ford Motor Company at this Symposium. We have, in the past year, undertaken a fairly ambitious program and what I would like to do this morning is to review the history over the past year; how we approached our particular problems, and what we feel we have learned that has general implications.

Figure 1 was prepared approximately a year ago. It represents the growth of technical and scientific computation at the Ford Motor Company over the past six years. Without going into the details of the particular equipment that is described, the main impact of this history has been that we have doubled our technical computer usage every eleven months for six years. That is an enormous growth. This phenomenal growth is one of the major reasons all of us are here today.

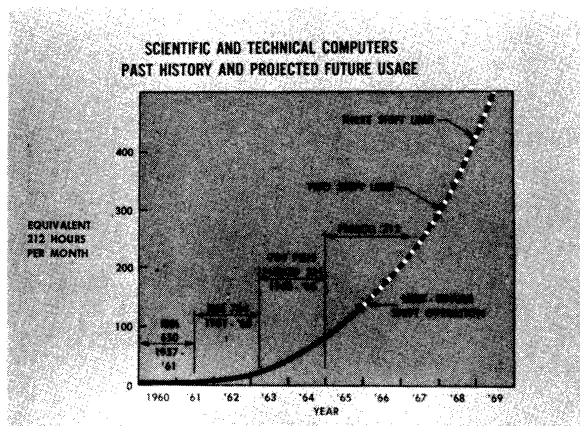


Figure 1

The Organizational Setting

The particular organization with which I am involved is the Technical Computer Services Department of the company. Our particular orientation is the scientific and technical computation for the company. This includes not just engineering but operations research and the applications of mathematical models to management problems as well. We have 70 remote terminals around Dearborn that are served "on-line" from the Center (Figure 2). We also have on-line satellite computers; GE 115 is the particular one we are using but it is typical of the UNIVAC 1005 or a 360/30 device that operates via a phone line. It is a small computer doing editing and small operations off-line, and yet can draw on the main facility for "on-line" computing power.

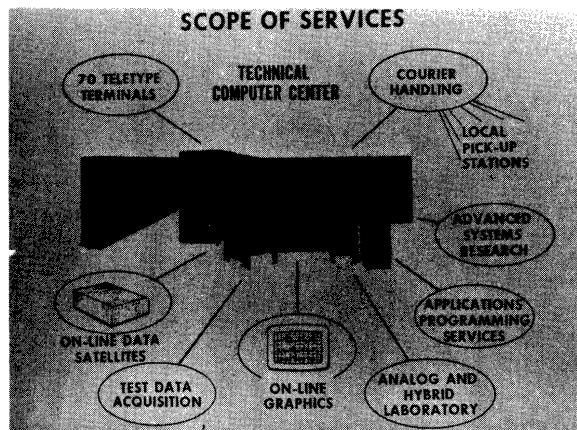


Figure 2

One of the key things about our operation is that within the Center we have both the applications programming services and an advanced systems research staff.

The programming at Ford is decentralized: We attempt to encourage our users to have their own programming staff. We serve about 200 organizations in the Ford Motor Company, its subsidiaries and divisions, and a few of our suppliers. Most of the programming is done by our using organizations. Where they have neither the staff nor the special skills, they can draw upon a central pool of people that are part of the Center. The advanced system staff does our advanced software work. We develop our own software for our facility, and this has allowed us to undertake programs that are rather ambitious for a user organization, even one as large as ours.

I would like to comment quickly on the fact that we are separate from the commercial data processing organization of the company. We operate under the engineering staff of the company and are oriented in a direction unique for a computing center. We are very much preoccupied with the notion that it is the user's resources we are trying to conserve and not the machine's. Indeed, in the problem solving setting, the key resource is the user's own labor and his time domain. The particular machine time he might consume, we feel, is a fairly modest part of the total cost. So we have organized ourselves administratively and technically, and also have established our software in such a way that we regard our measure of performance as turn-around time and the ease with which the user himself can solve his problem.

The Computing Center is a completely self-liquidating operation. We bill for all the services we render and obviously have to manage ourselves so we recover our costs. Nevertheless, the way we are organized is perhaps quite a departure from a setting in which you are processing a certain number of punched cards coming in and getting a certain number of linear feet of paper coming out with the efficiency of that operation defined in terms of the cost of doing a job. Approaching a problem that way you will come up with quite a different set of answers than we have.

Another important thing that I would like to emphasize in our particular organization is that we are very much oriented to creating alternatives rather than legislating standards. In a data processing setting where one is concerned with data control and standardizing languages and formats, one could perhaps come up with a different set of answers. In our particular setting, we feel that it is inappropriate for us to attempt to predict the usefulness of a particular technique to all classes of problems. As a result of that philosophy we have attempted to create as many alternatives as we can for the problem solver. We have available not only FORTRAN, but ALGOL, and MAD. People find these languages of differing utilities to different problems, and the creation of alternatives again point us in a direction that is perhaps different from most installations. Figure 3 shows some of the equipment we have at the Center. I will go into this later but basically our main frames are Philco equipment: Philco 212's. We also have GE equipment. The Center represents just under a dozen different computers that are linked together.

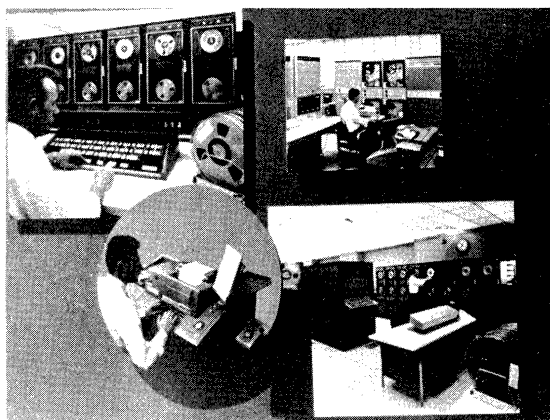


Figure 3

The Paradigm of Change

Let me start the narrative here by taking a look at ourselves, in a certain sense, in the fall of 1965. At that time we had a large-scale scientific processor, a Philco 212, which is a "6600 class" of machine (Figure 4). The Philco 1000 is the I/O processor. The classic mode of operation here was to bring your problems on punch cards - both the programs and the data - give them to the 1000 which would write them on magnetic tape, with perhaps 50 or 100 jobs on a reel of tape. The tape was then mounted on a 212 which would whip through those problems, writing an output tape. The output tape was again put on the 1000 and it printed the answers. Since the users have to wait for the whole batch to be done, they receive a turn-around time of, at best, a couple of hours. When things get pretty full there may be a wait of four or five hours. Fortunately at Ford we seldom have a situation where it is much longer than that. This idiom of usage is a pretty classic standard scientific job shop operation. Whether you are talking about a 7094 and 1401 or whatever other combinations, the pattern is a fairly standard cliché.

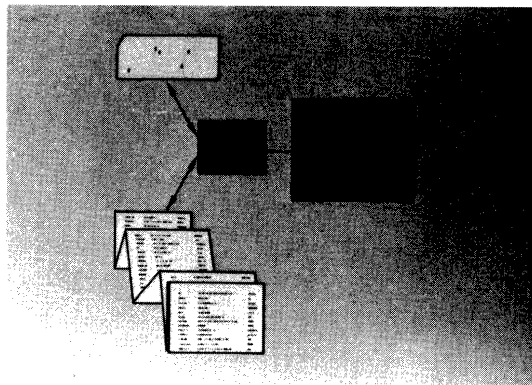


Figure 4

In the fall of 1965 we added a disk file. The main impact of this was to expand the available area of addressable memory for the 212 (about 200 times) because now we can put out about 42 million characters of information directly accessible by the 212 (Figure 5). This had a number of implications. The first one, of course, was that the kinds of jobs themselves went faster because the disk is a faster device to address than tape in some particular modes of usage. Also, and this is one of the important prerequisites to timesharing; it allowed us to put programs, system programs as well as user programs, directly on line so the 212 could call upon those any time.

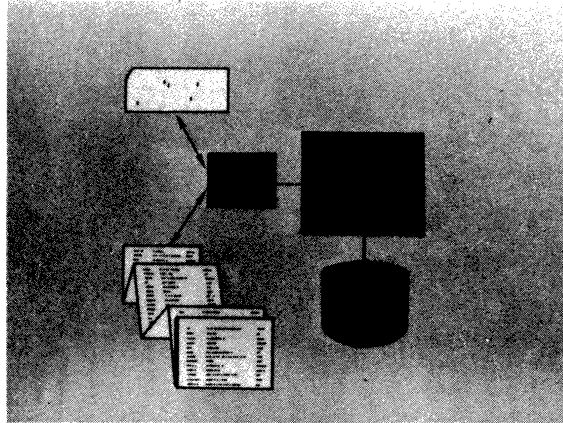


Figure 5

The next thing we did; and this brings us to about spring of this year, was to put in a complex of equipment that would allow remote terminals to talk directly via phone lines (Figure 6). It happens that this particular equipment will also do small jobs on its own. Where those jobs are larger it will call upon the 212. It is this experience and the impact on the company - that is, of installing an on-line facility which allows people at remote locations to directly harness the computer - that brings us here and that I would like to review.

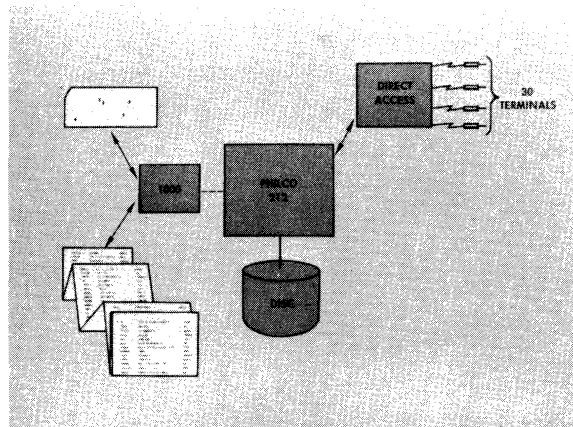


Figure 6



Figure 7

Figure 7 shows a Model 35 teletype which happens to be the unit that we have standardized upon. I was planning, (when I put the slides together) to sandwich a demonstration between what I just said and the coming presentation, but in view of the arrangements here I think I will finish the slides, and then demonstrate the facility.

In the time between February 1966 and August 1966 we had a phenomenal growth in the use of timesharing. This slide (Figure 8) gives a profile of our operations, in August 1966. We did 5,000 batch processing jobs. We went from about 3,000 jobs a month to about 6,000 jobs a month this year. These jobs run for about 233 using organizational units. In contrast to that, timesharing, which again I want to emphasize was introduced here only in February, is already doing 10,000 jobs a month.* That is twice what we are doing on the batch processing mode of operation for about the same user body through its 70 terminals. In August we had 3500 programs stored (now it is well over 5,000 programs).

OPERATIONS

AUGUST 1966

TIME SHARING

9,903 JOBS RUN
196 USER NUMBERS
70 TERMINALS
3,556 PROGRAMS STORED

BATCH PROCESSING

4,973 JOBS RUN
233 USERS
206 PROGRAMS

Figure 8

(*Editor's note: Author reports 20,000/month by year end!)

To put this in proper perspective the batch processing programs were written by our own applications programming staff of approximately 12 to 15 people. This staff set themselves the objective of accomplishing nominally about 10 programs a month. (Some of them were small, some very ambitious, but just as a way of measuring we set ourselves that objective.) At the end of 1965 this staff had, in fact, completed approximately 200 application programs or significant modifications to previous ones. We felt that we had more than accomplished our objective. Here we have, in approximately a seven month period, not 200 programs written, but close to 4000! Indeed, many of those programs are very small, some of them are scratch copies of something that later is going to replace it. The point is that if we had set ourselves the objective of writing 4000 programs, or putting it another way, 4000 programs were required to accomplish the kind of technical work that we had set for ourselves, there would not have been the programming labor available. This starts to give us some insight into one of the major impacts of timesharing. These programs are written by the user, for essentially three key reasons. One, the language is so simple a person can learn it very quickly. Second, because of the fact that the person is conversational with the machine and that the language permits certain ease of interaction, programs can be written very quickly and very simply. Third, you can write a tutorial package so that the computer will teach any person how to use itself in approximately 20 minutes. That is a minimum prerequisite. We have a whole syllabus of on-line programs that I will show you when I demonstrate the facility.

(The second half of this presentation presumes that you had seen a demonstration of our timesharing system.)

Basically what happens, and we will demonstrate this after the slides are over, is that you dial the computer, it answers, you literally say "Hello," and it asks you a few key questions like, "What is your charge number?" Having given the charge number, you literally converse with it in a language that creates the illusion of being almost natural English. Since the computer is participating with the user sitting at the terminal, if he makes an error in format or violates one of the syntactical rules, the computer tells him right away and assists him in getting the correction made. This makes it possible for the nonspecialist to use the machine directly.

Figure 9 profiles a sample of the jobs in August. We ran approximately 10,000 jobs on the timesharing facility. The vertical scale is the percent of teletype (or terminal) sign-ons. The horizontal scale is time expressed in minutes. What is significant here is that the median is approximately 10 minutes. That means that half of the people using the timesharing system signed on with the computer, expressed their problem, got an answer and were finished within a period of 10 minutes.

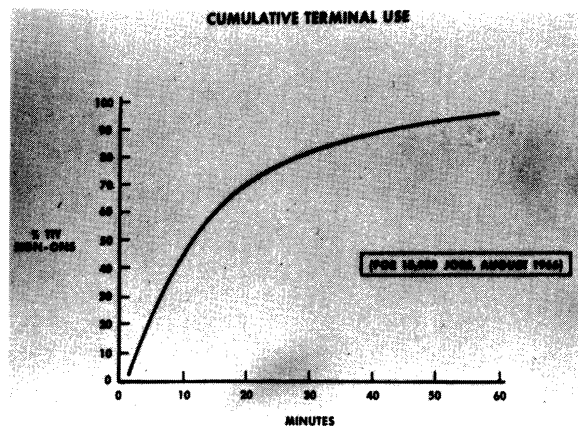


Figure 9

Others engaged in a half an hour to an hour conversation with the machine, developing some new program. Once the program is written it can be called up in a matter of seconds, the data entered, and the answer received.

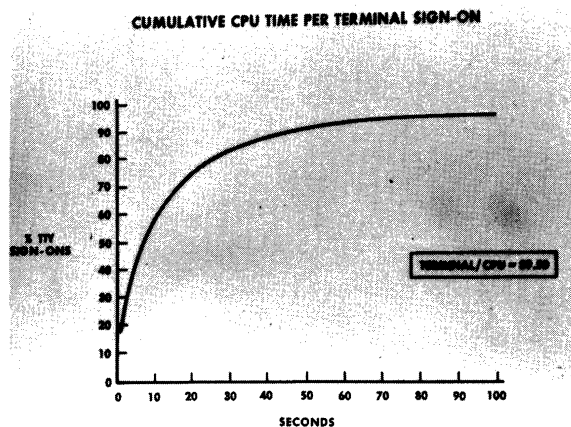


Figure 10

During the time that a conversation is taking place with the machine, you are not tying up the central processing unit. It is only when you actually "compute" that you are tying up the central processing unit. One of the key things of interest about a timesharing system is how much computer time one actually uses sitting there for an hour (Figure 10). For 10,000 jobs in August the terminal time to the CPU-time was about 40 to 1. One minute computer time would be like 40 minutes sitting at the terminal. The ratio varies from 10 or 20 to one to several hundred to one depending on how complicated your particular application is.

If you spend most of your time just conversing with your file you will use no computer time, that is, actual central processing unit time. If you are using very complicated equations you may be using a great deal of CPU-time. In terms of the percent of jobs, we find that about 80% use less than 20 seconds (or less than one dollar's worth) of computer time. That, in itself, is an interesting thing to assess.

Computer specialists can make a very interesting case that timesharing will never work. The reason it "could never be practical" is because the CPU is being operated at 60 to 80% efficiency. It is our feeling that this missed the whole point. The efficiency of the CPU is a fairly trivial part of the total package. Of the CPU costs for 80% of our jobs, (10,000 jobs from 200 different organizations) we find they use less than one dollar's worth of computer time! The time of the guy who is trying to get an answer to a problem, be it a brake design or a front suspension simulation or a economic reorder point or something of that nature, is a far more significant cost factor.

Notice that the users never see a "source deck" or an "object deck." All they see is the terminal. I would suspect that about 70% of the people who use our computers have never seen a punch card or a piece of magnetic tape. We have sort of a cliché in the center to the effect that punch cards and magnetic tape are useful but should never be seen outside a computing center. Anytime our user has to get involved with one of these, then somehow we have "failed." Obviously, we are not there yet. There is a great deal of work that is going on that definitely requires interaction of these devices, but as an objective, our notion is that people with a problem should be able to converse with the computer directly, get an answer back directly. All these other things are artificialities that should be relegated to the background and to a cadre of professionals who are behind the scenes and never act as middle men between the guy and his answer.

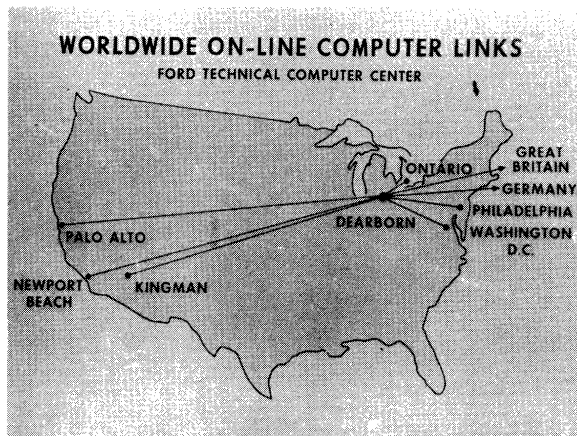


Figure 11

As a result of this ability to communicate with this computer by a telephone line, it quickly turns out that the Dearborn complex can be made available then to other Ford operations (Figure 11). These include Philco, which is a subsidiary, company operations on the West Coast, a test track at Kingman, Arizona, activities in Philadelphia, Washington, Great Britain and Germany and we are considering Australia. In the case of Great Britain our own message traffic is large enough to justify our own lease line. The business hours overlap about five hours out of the day. That leaves another 19 hours of non-prime time that are available to the British engineers and analysts who can dial the computer, about six hours a day every day on a routine basis, and use it conversationally the same way we will demonstrate this morning.

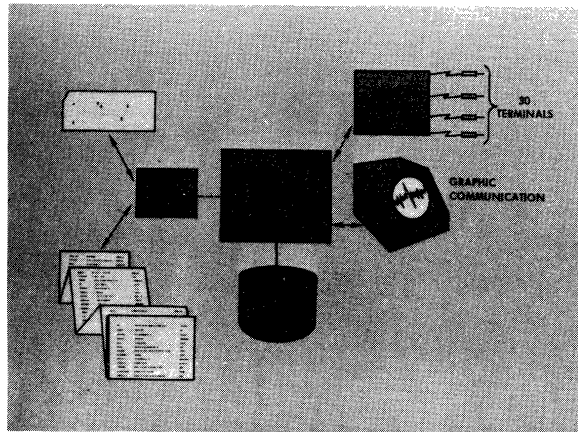


Figure 12

The next thing, of course, in terms of our growth will be to add graphic communications (Figure 12). I will not go into that today other than to say that if the natural world for the engineer is a world of geometric relationships then it is quite natural to pursue a geometric language to enable him to communicate with the computer. Indeed the teletype terminal, as useful as it is as a conversational device, is still basically algebraic and it is basically a stream of characters. There are many problems of great interest to us where the natural domain of the problem is a geometric relationship, whether it is a linkage or whether it is the graphical representation of a mathematical function, or a simulation of a dynamic system in which the linkages among the elements are much more conveniently expressed graphically than by some specialized algebraic language. Then, indeed, the potential of a graphic interaction is quite attractive and we are, as most of the companies in our industry, very aggressively pursuing this.

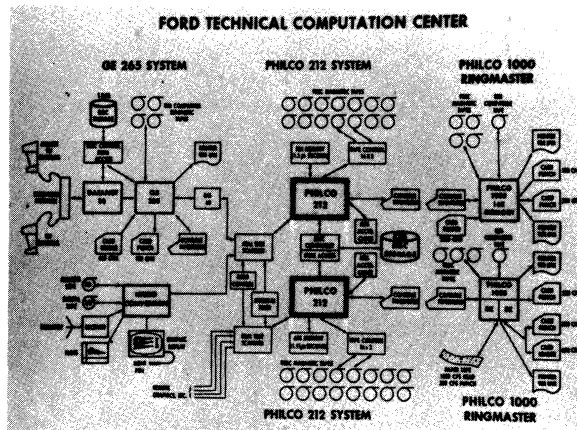


Figure 13

Equipment Configuration

I have talked, so far, fairly general and used some very simplified block diagrams. Figure 13 describes what we actually have on the floor. The primary frame in the Center is the Philco 212; we have a pair sharing the large Bryant Disk File. The Philco 212 executes instructions, from about one-half to roughly one-and-three-quarter million instructions per second. For comparison, take a Friden calculator and pace yourself at about five operations a minute. Operate one for fifty 40 hour weeks a year: you would have executed approximately one million operations, at the cost of one manyear to you. One million instructions on the Philco 212 will take approximately one second; probably more relevant, that one second costs you about 13 cents. We say with our tongue in our cheek, that we have analytical power at the Center at "13 cents a manyear." If you can get your problems into the machine where the machine can assist you with them, you can harness some tremendous economics. The real problem is getting it there.

Obviously, the punch card and the middle men that lie between the poor guy with the problem and his answer often does not contribute a great deal to getting his answer when he needs it. "Direct access" is the name of the game.

The facility in the upper left corner of the figure is the famous Dartmouth System. It is built with GE hardware. The software was designed and implemented by a group of undergraduates at Dartmouth College. The main elements of the system include a Datanet 30, a GE 235, and a rather unusual disk file built by Data Products Corporation. The remote teletype stations dial, via regular phone lines, into the Datanet 30. The Datanet 30 is, in many respects, a traditional computer except that it's orientated to do message switching: it will answer those phone

lines. Although fast, it is not fast as computers go. It is extremely fast in terms of the conversations that are being carried on with it. Its program permits a guy at a terminal to communicate with the disk file where he has his own private area dedicated. He can therefore establish a place to put his program or his data and commune with it. He can add to it, look at what is there, and modify it. When he wants to do some actual computing, the Datanet 30 will pass control to the GE 235, which is a medium scale business machine. The particular one we have is modified with an arithmetic unit, which gives it floating-point arithmetic which, in effect, means it can do modest scientific and floating-point calculations. The Datanet 30 schedules the entire operation but it passes control to the GE 235 which will pack up programs, execute them, allocating itself among those jobs which are actively computing at the moment. In other words, it will give each job a certain time allocation, set it aside and give the next job a certain allotment, set it aside, and continue this way "round-robin" among the active jobs. It will allocate the time slices, given any particular job, according to a procedure which attempts to create the illusion to all of them that they each have the computer to themselves. Ideally, the computer will do this in such a way that each job gets done faster than the input-output requirements would require. This illusion is not always perfect. There are times when it is obvious that one is sharing the machine with somebody else. However, there are just enough conversational things going on so that the power of this machine can support, for the class of problems one generally finds on a system like this, a very effective time-sharing service. There are some jobs that are too large for this complex of equipment. In fact, for the industry, the next major step is the evolution of large scale timesharing systems which we hope, optimistically, may be available in 1968. (More likely it looks like they will be a little more downstream!)

Most of the manufacturers, very candidly, were caught completely by surprise by timesharing because for several years there were a large number of professionals that were presenting some very convincing arguments that such systems would never really be practical. They maintained that such systems were strictly laboratory novelties that would be confined to the economics of a government research environment. However, it is clear that trends are stronger than the industry anticipated and, as a result, all the major manufacturers are presently scrambling to adapt equipment that will allow conversational convenience with large scale power. Indeed, we are moving as rapidly as we can in that direction.

The Philco Interface

At Ford, as a interim measure, we have done something a little different. We do have these two 212's; computing power that is adequate to do most of the jobs that we encounter. What we have done is to design

some equipment ourselves, which permits one computer to call upon the other computer for help. If there is a job that is requested by the terminal via the Datanet 30 and the GE disk storage that exceeds the capability of the 235, it will interrupt the 212, which is presumably doing some background batch processing, and pass that job onto the Bryant Disk. After it has made the transfer the 212 will finish what it was doing to some appropriate break point, pick the on-line job up as a preemptory task, accomplish it, and pass the answer back to the GE system. There are some options depending on what the program asks it to do. It can write answers out on magnetic tape for ultimate dumping on a regular line printer or data plotter, or the usual data resources available to a computer. Or it can take small parcels of those answers, like five minutes of teletyping apiece, and pass them back to the terminal. There are some restrictions but this means that the teletype terminal user now can call upon the 212 with its complete library and richness of languages to solve very large problems. He can get those answers out in detail (via mail) if he wants to. He can get a quick look while it is happening if he wants to. It represents an interim step to the day which we are actively moving toward in which the large scale machine can be fully timeshared.

Analog Data Facilities

We also have a hybrid conversion facility, the Ambilog 200, which allows analog data such as track or laboratory test data, to be operated on in analog or digital fashion. It can prepare the material for detail calculation with the Philco 212. It turns out, because of the hybrid nature of our device, to be very convenient to put a display terminal on it with a light pen and actually observe what is happening. If there is a recording of interest you can see the wave shape on the graphic display terminal and, by means of English words in the margin, you can point the light pen and execute mathematical transforms on that wave shape, such as a power spectral analysis. The interesting thing here is that the engineer, at the display terminal, does not have to know about programming. In effect, he can call any of the large number of techniques by simply pointing the light pen at an English word referring to the one that he wants to use.

The 212 is actually organized for military command and control applications and, as a result, it lends itself very nicely to linking up a large complex of equipment like this.

Software: A Perspective

I would like to take the opportunity now to comment, if I might, on the use of the computer in terms of the "software". So far we have talked about hardware and most people, when they think about the computer, they think about the part that they can see. As they walk through a center they see the boxes: the 235, the Datanet 30's or the 212. We read constantly about IBM, GE, Burroughs and RCA coming out with their new boxes and I think one of the things that is lost, and which needs to be emphasized, is that these mean relatively little to the computer user.

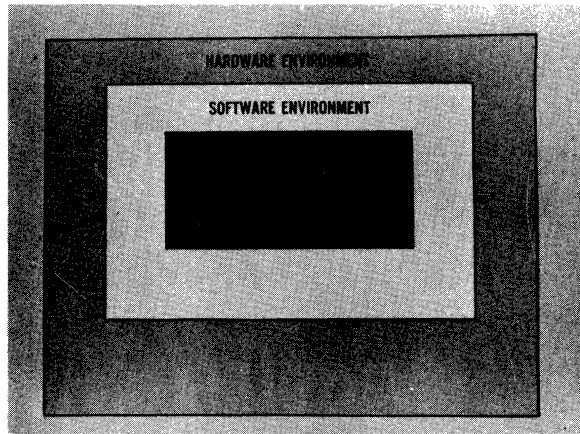


Figure 14

As a user if you have something you want the computer to do, you describe this by means of what we call a "computer program". This computer program is usually written in some language which is, hopefully, as convenient as possible for the user. It can be almost like English. It essentially represents a set of instructions to the computer that can be very explicit and, in some cases, surprisingly general. But that user-program is not written in terms the machine itself can use. It turns out that there are a set of "superprograms," if you will, that lie between your program and that hardware (Figure 14). For lack of another word, and since this superprogram is a part of the facility, we have seen the emergence of the term, "software," where the hardware and the software both are part of the computational facility. Many people use these terms very generally to mean any kind of computer program. In the particular sense I am using, it means those superprograms that are parts of the facility that allow the user to come to that facility with a program expressed in a language of his convenience. It essentially accomplishes what the user wants.

Now my point is to indicate that the user does not really care what the hardware is. The user who has a FORTRAN IV program, FORTRAN IV being a particular language, does not really care whether it is an RCA, IBM, GE, Philco, or whatever kind of computer at hand. What he does care about is the effectiveness of the available compiler. Is it any good? Will it get this program executed? How much will it cost him? These are the answers that he is after. This approach or attitude is exemplified in the timesharing. When we dial the terminal and talk to a computing center, as we will be doing quite a few times today, we will be talking to various kinds of equipment. The important thing is that the guy sitting in his chair at the terminal does not know or care what equipment is there. He does care what kind of software environment has been provided. Is it special purpose or is it general? What are its constraints? Can the program that he puts in be 2000 characters, 6000 characters, or 100,000 characters? What resources are there?

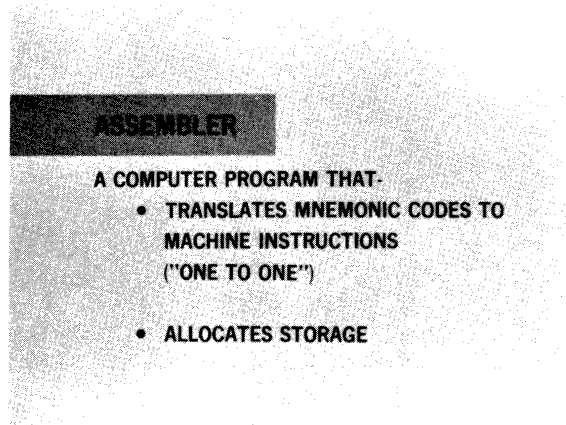


Figure 15

Obviously, what the software can do for him is a function of the hardware it resides in, but the point is that what is there, whether it is a Bryant Disk File or a Data Products Disk File, it's of little concern to him. The hardware will determine the efficiency or, I should say, the potential efficiency. But the user does not care and this is important, because it means that there is potentially a completely different structure of our industry. The guys using the computer may have very little need to get involved in the intricacies here. In terms of costs, years ago about 80% of your dollar would go for hardware and a fairly modest amount of your dollar, 20% or so, would go for software, (very pedestrian software: what we call assembly level language.) Presently about 50% of your dollar goes for software and according to the recent industry survey in the Electronic News, it is projected that by 1970 the ratio will be 80:20. This means that for every dollar spent on a computer, about 80 cents of that will be a software investment rather than a hardware investment.

One of the first things that came along when the computers were first built in 1950 was the assembler (Figure 15). This was simply a program where you had to know how the machine operated. Essentially you had a one-to-one translation of what you wanted that computer to do. The remarkable thing about a computer like the 212 which I mentioned before is that it is capable of executing about a million instructions per second: that is, in one second the machine executed one million instructions that had been explicitly defined by someone or some process. The speed is not the only amazing thing: it is more the fact that a program can require billions of steps explicitly described.

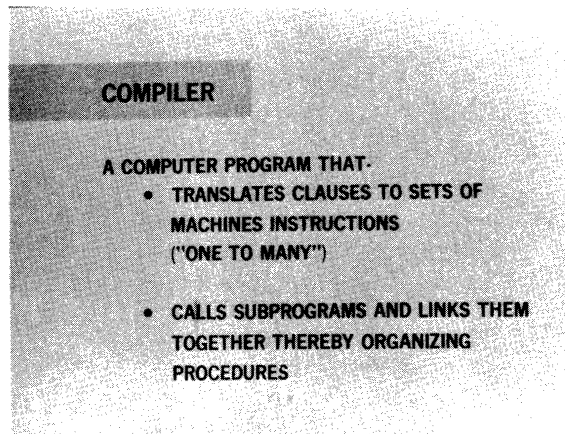


Figure 16

One of the major revolutions that came along in the software industry was the concept of a compiler (Figure 16). This is something that will take a higher language and translate it into terms a machine can use. One of the key implications here is that now we can express a statement that will generate many computer instructions. If you want to execute a certain mathematical function the compiler will simply take it and assemble all the instructions it will need to execute the mathematical function the way you want it. It suddenly gave the programmer tremendous power. The compilers, of course, were the first major breakthrough in using the computer practically.

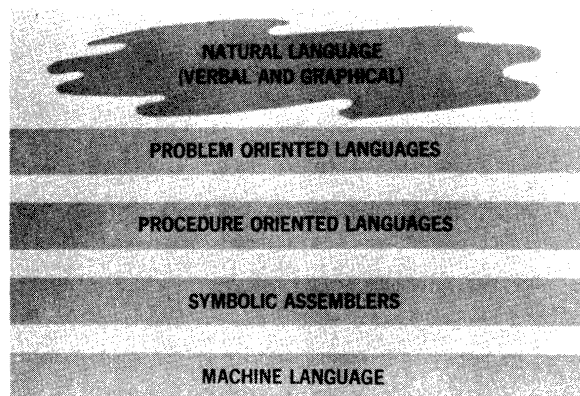


Figure 17

In the early 1950's we talked in machine languages; then came the symbolic assemblers, and then the procedure-oriented language (Figure 17). I am talking about COBOL, FORTRAN, ALGOL, and MAD - those languages in which you must understand the procedure in detail. The next step is the "problem-oriented" language. If you are operating in a mechanical engineering environment, you can express your problem as a network of springs, masses, and dampers. The differential equations implied can be formulated by the machine itself and you can direct dynamic

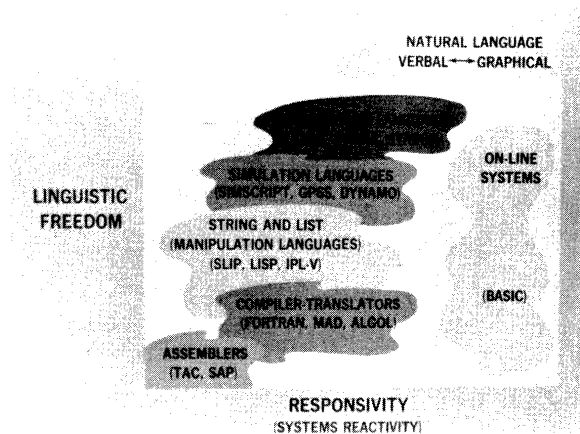


Figure 18

simulation without even going through the procedures. What we are doing is growing towards a natural language communication machine. By natural language I do not mean just verbal: I am also thinking of graphs.

To elaborate on this a little more refer to Figure 18. If we attempt to plot linguistic freedom, that is, the degree of freedom of the programmer, in one dimension and the responsiveness of the total facility in the other, we see ourselves going in a direction away from the assembly level language. This is known only to the systems programmers these days. We have gone a step higher with a string of enlistment manipulation languages such as SLIP or LISP and through special engineering languages; we are moving towards giving the user more freedom with less computer training required as a result, and also making the whole facility more responsive. This attempts to explain why a language like Dartmouth Basic, which is extremely simple, has had such a profound impact on our business. It is a conversational language and you can express your problem literally in minutes instead of spending two weeks going through code edits and can get your answer right away.

The Future

We are moving toward the day when we can sit at a terminal and converse with the machine in a language which is essentially to the user, his natural language. It does not mean a completely free format of English in the usual sense, although it might. It simply means a language in which any restrictions it has are made completely transparent to him. If he encounters problems, it helps him. If he forgets to state an initial condition, it questions him, so that if he is looking at some data and it has some particular shape, it shows him what it will be as well as giving him the polynomial. If he wants to make some adjustments, he can do it.

The implication in terms of hardware is that speeds are becoming phenomenally fast. Breakthroughs in speed may not be as great in the future as they have in the past because we are getting to the point where the speed of light is limiting us. Memory is getting extremely cheap and what is just as important, they are also becoming very indirect. Our worries about dealing with them directly are being delegated to the software and we are near the day where our major concern is to create a memory that has essentially an infinite dimension to the programmer. This could have phenomenal implications because the power of the computers can be geometric with the memory size.

In terms of software, obviously we are getting more communication oriented, direct-access oriented. We have the combination of the natural languages along with the ability to create the apparently infinite memory. It is going to completely change the concerns we have had in the past where we have been trying to figure out some way to get our big problem in that little machine.

The Information Utility

As the computer gets larger, while its power increases geometrically, its costs go up arithmetically. This leads us to the main implication of our experience at Ford: the information and the utility. If you buy a large machine you usually cannot afford to keep the thing busy continuously; thus your costs are excessive. If a particular organization like a small business or a sub-activity of a medium scale business could have a large memory during the short periods of time they may need it then they can harness the efficiency of the large scale system with the cost of just what they use.

We will see later today that there are organizations equipped to offer services which allow a person to dial in and get just the particular kind of service he needs. This is analogous to the power industry. In the early days of the industrial revolution, one could use steam engines to harness the power of the machine to do much physical work. But since they were only effective in large sizes you found them only where there were large jobs to be done. With the advent of electric power came the ability to distribute that energy and one could draw as little or as much power as he wanted where the job happened to be. This, in effect, is what is happening with the timesharing facility. If someone in the offices at Ford has a little problem, he can dial a computer and get his answer using just what he needs. On the other hand, if he needs a large simulation he can get that also. He gets charged for that portion of the resources he has actually used and this is one of the key notions of the utility concept.

There is a second more provocative point about the utility, and that is the issue of software. It is becoming more and more apparent to the professionals in the field that the whole issue of computer facilities is a software story. One of the most frustrating things to plague industry ever since the computer industry got started is this issue of software development and the economic support under the development of software. The problem is very simple. If you develop the software there is no way to protect it. Patents do not work and such an approach is frail. If someone gets a copy of your software there is nothing to prevent them from reproducing it. It is just not the kind of thing that lends itself to protection, and unless you can get it used by many people you cannot amortize the cost of developing it in the first place. Essentially software today is developed one of two ways: either it is supplied by the manufacturer with his facility, (and history here has been fairly painful with all of them) or you end up having a consultant do it for you. This means you end up tailoring your own software and bearing the cost of it yourself or you end up with compromising what you really want.

The computer utility offers, finally, a very provocative answer to this. Let us assume that John Doe has developed a particular way of solving a certain kind of problem that is very outstanding. He can store a program, in the commercial utility and make it available to the users on a rental basis. A user of this program would not only pay for the computer time but also an additional cost for the use of that particular program. Yet since it resides in the utility, means can be provided so that it may not be copied, or compromised. This means that the one who put the program there can find a source of revenue for his development. At the same time he can be free of the fear of compromise because it remains within the utility. The executive system can allow its use but not its compromise. So as a result, there is an economic base, under which people with skills in these areas can invest in the development of languages or algorithms or whatever might be useful software, and have a way of recovering their cost. This means that the whole complexion of the industry may change. It may be the key part of the industry five or ten years hence, where essentially its resources are not just the speed of the machines, or the cost of its memory and this sort of thing. What we have is essentially a software brokerage concept, and it is one that, to us, seems to be one of the most interesting solutions to the key problem in the computer industry.

I would like to comment briefly before we go to the demonstration that Ford's orientation in this particular facility is the problem-solving use of computers. Quite often we hear, "Timesharing is great for engineers and scientists but my problems are data processing." I would like to say two things. One, it is clear that these distinctions are

disappearing. It is interesting to note that the really large machines today are doing the large problems - business problems, not engineering problems. The problems of the Philco 212 at the Ford Scientific Technical Computing Center represents billions of instructions executed. Our problems in linear programming, marketing distribution studies, allocation models of various kinds, are very complicated mathematical models. The classic dichotomy between business and scientific problems has become an anachronism. I think one of the most interesting things in this symposium on timesharing, is the utility that is going to be following me, to be presented by Charlie Adams, on data processing. It is a commercial-user-oriented, not a scientific facility, and I think that speaks more loudly than anything I can say. At the Fall Joint Computer Conference a few weeks ago, the controversy still raged; people still have not seen the message.

The last thing, put in loosely because I assume there will be some ample discussion, is that we find many, many anthropologists and others suggesting that the computer implies centralization. These big machines will cause business enterprises and organizations of all kinds to become more centralized. And indeed, this may be the case, but there is a contrary view. With timesharing it just might be that we have the tool, for the first time in man's history, to decentralize an organization. If, indeed, we can make the storage - the memory of that computer - a medium of communication between the various elements of an organization, a participative media, that will insure that if there is someone who needs to know about some particular fact that the right people know and the right adjustments are made - if that is true, then maybe for the first time in the history of business we have the tool which is really required to fully decentralize an organization. I suggest that this might be an interesting thing to consider when we get into the discussion period. Now at this point I want to demonstrate the Ford facility. I should hasten to add, we have two peak loads during each day. They occur essentially in midmorning and midafternoon. There is a slight dip during the lunch period. Unfortunately, we are scheduled right at the peak time so I will be shocked if we get through the first time.

(Here followed a demonstration of the facilities at the Ford Computing Center.)

DISCUSSION

- Question: Do you have a computer at Ford's where you take analog information directly into a time-shared computer and get information back in an on-line sense?
- Mr. Missler: The application that comes to mind first is our test cells. We have engine dynamometer test cells. The satellite computer, if you will, that is doing the on-line data acquisition will be able to tap the timesharing system for processing that it prefers to delegate to the timesharing system. The timesharing system is accessible, and in this case we plan to use phone lines. Anything that you want to put out there at the end of the line that is programmable. We have a GE-115 over in Body Engineering. Their world happens to be punched cards. They do coordinatograph work and they have a large body of punched cards which require editing before they go into the big machine. It turns out to be convenient to put those into a satellite computer unit and to do some preliminary work before passing it to the big machine to process. We have a programmable device at the test site to do those things directly that you want done immediately. They do have the capability to pass a message up the line to the central facility. There are many people who will say we do not have the necessary bandwidth in a voice grade line, but I think that there are resources in coding theory that suggest that we can very much more effectively use the bandwidth of a phone line than we are now doing. This goes for a teletype terminal type of thing as well as the analog applications. We could talk more about this at lunch if you would be interested.
- Question: Is there a traffic problem? Is there a need to restrain the amount of time that an individual will stay at the keyboard?
- Mr. Missler: We have been very sensitive about not playing policeman at the Center in any sense. We leave this up to the individual using organizations to police. The obvious answer is to put excessive loads in another terminal. And we are very quick to sign them up if they need it. There are enough terminals around the Ford engineering facilities that if you cannot wait for someone to finish, you can probably walk around to find one that is free. An interesting statistic that may be of general interest here is that we have 30 telephone channels into the Datanet 30. It actually could take 40 but the system performance turns out to get pretty bad over 30, so we just limited it to 30

lines. At one time we had 30 teletypes installed during our period of growth and at no time at which we only had 30 teletypes did we ever have more than 10 to 15 terminals actually on the facility at any one instant. In fact, one morning while working with General Electric, we thought we had better make a full load check. So we called up everybody, all the 30 terminals, and said, "At 10:15 everybody go on the air." Even when we did this we did not get them all on. It just turns out statistically that they are just not all on at one instant. There is always somebody scratching his head, typing something else, signing on, signing off, or something. Right now we have approximately 70 teletypes in our subscription community and as you noticed this morning, much to my surprise, we got in the first time, we did not get a busy signal which usually happens. For those of you that are operations research oriented, the interarrival time is less than a minute, which means that the time for someone to sign off and the next fellow to sign in is less than a minute!

Norman Scott:

Mr. Missler has just described to us a very interesting setup directed primarily toward engineering and scientific computation totally "in-house," that is, totally within the confines of one corporation. Now, if you do not happen to be the Ford Motor Company or an organization of equivalent size, you still have the possibility of tying into a computer utility, a service provided commercially. Our next speaker is Mr. Charles W. Adams, who runs one such service and who will tell us something about the concepts of this kind of provision of computer service.

Charlie Adams has been around the computer business for almost as long as there have been computers. He did his undergraduate work at MIT and got his degree in physics there in 1948. In 1949 he received a Master's degree in mathematics. At the same time he was working in the digital computer laboratory at MIT on the Whirlwind computer that I referred to earlier this morning. He was one of the first to become interested in the use of the computer in a business data processing sense. In 1952 he conducted a two week summer course at MIT in this general area and wrote what I think is the first and still one of the best texts on the use of the computer in business data processing.

In 1955 he left MIT and took a year's leave of absence with the Westinghouse Electric Corporation where he was the electronic data processing advisor to the Director of Office Methods. In 1956 he left MIT and became the EDP advisor to the controller of the Creole Petroleum Company in Caracas, Venezuela. In 1959 he returned to this country and established Charles W. Adams, Associates which has been a consulting service for people who wanted to get into the computer application area. At present he is also President of the Keydata Corporation which is a subsidiary of the Charles W. Adams, Associates Company. The Keydata Corporation is the organization that I referred to as a computer utility and is the one which Mr. Adams will describe to us today.

THE COMPUTER UTILITY - ON-LINE COMPUTER SERVICE AT RETAIL

Charles W. Adams

President of
Charles W. Adams Associates, Inc.
and the
Keydata Corporation

THE COMPUTER UTILITY - ON-LINE COMPUTER SERVICE AT RETAIL

Thank you very much, Norman. Good Morning. Perhaps you are aware that the particular computer utility with which I am most immediately concerned and familiar is one which is devoting the bulk of its efforts to providing services to business users. In the afternoon you are going to hear a talk on liberating the engineering user from the business user. I hope and trust that one of the theses of that talk will be that the advent of the remote console time-shared computer facility will greatly facilitate easy computer access for both engineering and business users. This can be done using two separate operations as Ford, needing a large facility for each, has done or using a single computer in which the people processing data and the engineers work together, each without knowing that the others are using it. For, as you have seen from Mr. Missler's demonstration, the essence of the remote access system is that the user appears to have immediate personal access to a machine of his own. The system is either improperly designed or overloaded if it significantly impresses on any user the fact that other people are using it.

The very fine presentation of what is going on at Ford and the impressive demonstration that you have seen, must lead you to a lot of ideas about what could be done with the remote access system to any engineering department or other parts of any company. You have been told that the scheduling algorithm is one of the things that time-sharing specialists discuss a good deal. Mr. Missler said that some think it is the essence of the design problem; others disregard it entirely. I think perhaps the reason that people have put emphasis on the scheduling algorithm and the response time is because then they are conversing with their file, they want and expect very fast response. They are in a hurry to go on. In fact, some of the systems that are in operation give you a choice of whether you want lots of chatty English coming back to you or something less chatty. (One of the very early systems that our company's co-founders helped develop back in 1958 had a command call "BE BRIEF" to be given to the computer when you no longer wanted to be chatty. You were still conversing, but doing it Walter Winchell style.)

The scheduling algorithm should be such that the response to conversational inputs such as are involved in editing your own program seem essentially instantaneous. That does not necessarily mean that after you have a program ready and you say RUN that you have any right to expect that it will be run instantaneously. Even though it might require only a quarter of a second on a Philco 212, that run is, after all, the thing you have been leading up to. If it takes five seconds

or even five minutes for it to run, it should not matter. The point is that the response to a RUN command can and should be quite different from the response to the input of another statement in the program. You can afford delays in the one case where you can not in the other. One thing you cannot afford in the event of a long run, however, is the feeling that you may be out of touch. Even if you are going to wait five, ten or fifteen minutes for a long computation when the computer is heavily loaded, the system ought to provide you with a periodic report that something is going on. Even a simple little "chung" sound out of the teleprinter from time to time (e.g. figure shift, letter shift) can be used to give you assurance that you have not become disconnected.

Another variable in time-shared system design is concerned with the question of how the user communicates with the computer and to what degree he must be able to communicate his problem as well as his data. There are many different ways this can be approached and I do not think that any one of them is the ultimate answer to every question. Presumably the ideal thing would be to provide a mix of capabilities to meet differing requirements. To illustrate, I would like to spend a few minutes describing what KEYDATA Corporation is doing for small business users and pointing out some of the differences in design to which this has led us.

The system that you have seen demonstrated, which is more sophisticated than, but relatively typical of, the bulk of the systems being offered commercially to engineering users today, is characterized by remote Teletypewriters which are connected into the computer by dialing when access is wanted. It is programmed by the user in any of several programming languages and ordinarily uses the paper in the Teletypewriter as if it were scratch paper. It is characterized, usually, by what is called a "half-duplex" connection to the computer in which everything typed on the keyboard is printed on the printer and everything the computer sends back is also printed (which incidentally leads to some little difficulty if computer and user both decide to print at the same time).

There is another way of connecting Teletypewriters - a trifle more expensive but somewhat more attractive in certain cases, particularly in business applications. This so-called "full-duplex" connection is used in the KEYDATA Station (KDS) which KEYDATA supplies to its subscribers. It differs in that the keyboard is connected to the computer but not to the printer. Everything that is printed comes from the computer. While a half-duplex logic can be simulated when desired by having the computer immediately transmit back every character that it received, the KEYDATA subscriber's operator invariably types blind.

She sees nothing when the keys are struck, only the response which comes back after she strikes the "end of message" key. Do you ask, "Would that be a good thing to do?" Do we say to ourselves, "I would not like that, would I?" No, because you and I (if you are like me) are hunt-and-peck artists. We like to know where we are as we put each character in. We often make mistakes, as Mr. Missler has aptly demonstrated. As we have seen, these are easy to correct as far as the computer is concerned, but you have already spoiled the appearance of the typewritten sheet and cannot easily undo your handiwork. If you want to produce a piece of paper that you can give to a customer of yours (for example, an invoice), you do not want the form spoiled if you can possibly help it. The full-duplex logic used by KEYDATA substantially aids in this.

Unfortunately, I do not have a remote demonstration set up for you, but I did bring a film along. Some small portions of it you will find a little dull if your orientation is terribly anti-accounting, but I think there are several messages we can get out of it. (Film shown here)

The film you have seen is, of course, a pitch for KEYDATA's kind of service. I am sure Dean Scott picked me to give this talk partly because KEYDATA's current commercial interests are far out of your field and geographic area - we have no immediate axe to grind either with engineering users or with the State of Michigan. The service that you saw in the film is being provided to a couple of dozen organizations in the New England area now, with a number of others waiting to go on. The fact that they are still waiting and not yet on, unfortunately, would seem to support the widely-held theory that says "let the user do his own programming." In other words, KEYDATA undertakes to do the programming for its subscribers. Once we get the hang of it, this should require only a limited amount of specializing of generalized programs already available and should save the subscriber money and make more efficient use of the computer as well.

While this approach naturally means that the ability to put people on is restricted by the programming capability available "in-house", one of the appealing aspects of KEYDATA's service to the business user is that it provides an answer to a problem rather than merely a tool for solving the problem. This is usually much more attractive than having IBM or one of the other manufacturers bring in a piece of equipment which the customer must then staff up to program, operate and generally concern himself.

I have already commented on the full-duplex operation you noted in the film. Not having immediate printout improves operator speed and reduces the spoilage of forms. The operator who needs to see what she is keying as she keys it, is not a very good operator.

The operators also benefit greatly from KEYDATA's keyboard layout which permits one-handed numeric keying much as on a keypunch or an adding machine. Another advantage is that the operator can go at her speed and let the printer go at its. Since teleprinters are fairly slow and the amount of data in is only a small fraction of what comes out (in our case about one-sixth), many of our subscribers' operators key ahead enough to be able to take a coffee break for free.

As an aside, I might remark that we do not encourage this, partly because there are difficulties if the forms get out of whack while the station is running unattended. Usually, after one or two disasters, the operators do stop going far ahead. In the beginning, we had another problem - some users would, as the 7:00 p.m. shut-down approached, key as rapidly as they could and build up 30 or 40 minutes of printing. The way our system is set up, we were faced with the dilemma of either shutting the system off as scheduled (thereby dumping output that they needed) or letting it stay up to serve only these one or two persistent users most extravagantly and at no extra charge. As you can see, there are a wide variety of problems in operating a computer utility.

Figure 1, a configuration chart on a conceptual basis, shows what is in the KEYDATA system. The fact that the Teletypewriters communicate with small computers (as they do in the Ford system) is attractive because these can be put in remote locations and can communicate over single voice-grade lines with the larger main frame. It would be possible, for example, to serve 40 Teletypewriters (or KEYDATA Stations, as we call them) in the New York area from our center in Cambridge with only one phone line running from Cambridge to New York.

The facilities shown here are physically quite similar to what is in the Ford system, but the emphasis is different: files are the main item from a business processing point of view, computing is almost incidental. We deal with much smaller records but make many more accesses than is likely to be the case with the typical engineering user. He calls up quite large records, brings them onto a drum, then accesses them from there. Here all of the files are broken down into individual records, describing individual records, describing individual customers or stock items.

The economics of file storage being what they are, it becomes important to put as much of the file as possible into a conventional batch processing system. The periodic reports that we deliver are produced from files kept on magnetic tape and run using batch processing techniques. This has a number of economic advantages, including the fact that one cannot, at the moment, economically put a fast printer

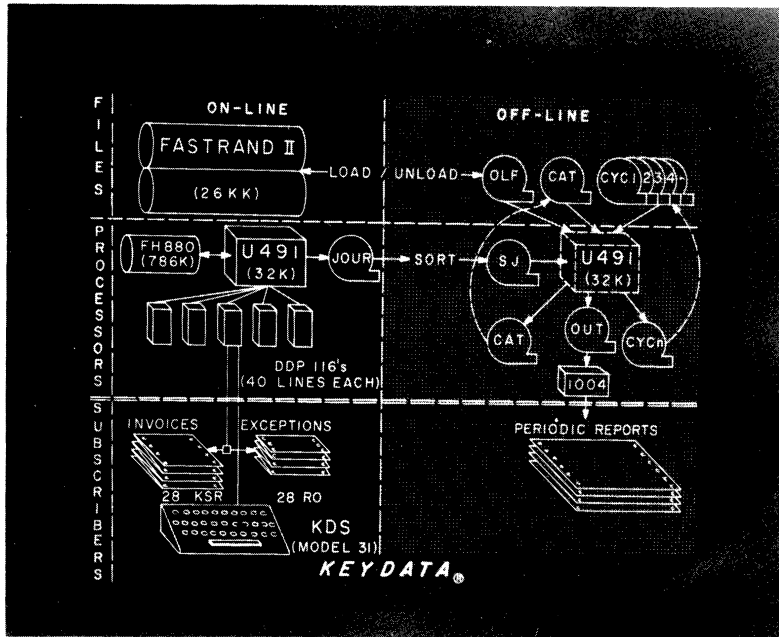


Figure 1

TYPICAL MONTHLY CHARGES		
MODEL 31 KEYDATA STATION		
(MODEL 28 KSR & RO TELETYPEWRITERS ON A SINGLE FULL-DUPLEX LINE.)		\$ 160
(ADD \$2.34/MILE FROM CAMBRIDGE)		
FASTRAND FILE STORAGE		
500,000 CHARACTERS @ 50¢/THOUSAND	250	
ON-LINE INVOICING		
2,500 INVOICES @ 7.5¢		
12,500 LINE ITEMS @ 2.5¢		
4,000 POSTINGS & INQUIRIES @ 2.5¢	600	
OFF-LINE REPORTS		
30,000 LINES @ \$3.00/THOUSAND	90	
		\$1100

Figure 2

on the premises of small subscribers. Producing the amount of printing required to prepare statements for each subscriber to send out to all his customers at the end of the month would be more or less prohibitive on teleprinters or low speed printer type equipment, so that going to line printers is a relatively important thing.

In any utility, the question of pricing is, of course, a crucial one. Here again, the approach used by KEYDATA in dealing with business subscribers is quite different than that used by organizations currently offering services built around the BASIC, JOSS or QUICKTRAN languages. Figure 2 summarizes briefly KEYDATA's approach to charging subscribers. You notice that the charges in the top half are by the month. They include the cost of the Teletypewriters and the telephone line, the latter being leased or dedicated lines (rather than dialed connections) because all our users tend to use machines much more than half of the working day, many of them almost full time. Many have more than one KEYDATA Station in use because they cannot handle all of their processing on one. While we can and have, in demonstrations, operated on dial-up connections, we do not presently accept subscribers on a dial-up basis.

You noted in the film and in the configuration diagram that we typically use two printers on the end of a single teletypeline. Thus we can have two forms going at the same time and produce invoices and exception notices on two different preprinted forms without having to change the forms in the Teleprinter. This second printer, a receive-only Teletypewriter, is selected by what in telephone jargon is called a "stunt box." The computer decides which printer to use and instructs the station accordingly. The Model 31 KDS in the diagram is therefore merely a Model 28 KSR and a Model 28 RO on the same line, with minor changes in keytops and the print box.

In addition to monthly charges for the KDS, we charge by the month for the file storage used and by the transaction for both on-line service and off-line service. You will note that there is no reference made to milliseconds, microseconds, dollars per second, or any such thing; rather, every line of invoicing that is produced rings up $2\frac{1}{2}$ cents and every invoice (exclusive of the line items on it) rings up $7\frac{1}{2}$ cents on the transaction-charge meter. Other transactions ring other unit charges depending on the application involved. The subscriber who has a very complex processing job pays a little more for each charge; if his requirements are particularly simple, he pays a little less. The charges actually stem largely from cost of the file access rather than processing and do not vary much from the $2\frac{1}{2}$ cent rate. In effect, we have adopted the telephone-company approach of charging by the message unit rather than attempting to charge either directly by machine time used or merely by connection time.

Of the several conflicting theories on pricing for engineering users, I personally think the best is the one in use at Ford in which you pay by both the connection hour and the processor second. Another common approach is to charge by the connection hour only. It may sound to you, as it does to me, a little ridiculous to pay the same for being connected whether you are putting a demand on the central processor or not, but there are very strong advocates of this pricing policy. They argue that in using a long distance telephone connection from here to California, for example, you pay by the number of minutes you are connected whether you choose to talk or not. While this is of course true, it is not the crux of the matter. When you are connected on a long distance call in the United States you are generally tying up the telephone company's facility whether you are talking or not. If you are talking trans-Pacific or trans-Atlantic, where various forms of time-division multiplexing are used, you do not tie up as much facility if you keep your mouth shut as if you chatter. In this case, you really should be paying more for the time you are talking, just as in a time-shared computer where you only put a significant load on the system when you ask it to compute.

Since you do put some load on the computer merely by being connected, you ought to pay some connection charge, though there are even some organizations who charge only for the amount of processor time used and not for the connection time. I suspect that when an industry standard evolves, it will boil down to paying by the month for the Teletypewriters, by the connection hour for the time you are connected, by the transaction for file access and by the microsecond or the memory cycle for the processing demanded. If you try to avoid pricing all of those components separately, you build in loop holes so that some people will be able to make much better use of them than others.

There are people who say, "Well, if you charge me by the microsecond and I sit down at the computer to do something, I do not know how much it is going to cost me." True. There ought to be some way of limiting the rate at which charges build up to protect the over-zealous or inexperienced user from himself. A well-designed system would allow him to establish a limit if he wished, so that when he starts making processing demands that are higher than he anticipated, the computer response slows down. A further argument against charging only by the console hour is that if you sit down at the console at 10 o'clock in the morning and do a given job, then sit down at 10 o'clock at night and do the same job, paying for both by the console hour, you will do the one a lot cheaper than the other. The charge becomes less, rather than more, predictable because the amount of console time that it will take you to do the job depends upon how loaded the system is. Offsetting this, you would perhaps expect to get better service when during off-peak hours, and probably the prices should be different at different times, but on a planned rather than a chance basis.

I hope that in these ramblings I have conveyed to you that there is no single accepted pricing philosophy today. Nor is there likely to be one in the near future. The ones that are now very simple, I think, are going to get more complicated and the ones that are fairly complicated may get a little simpler as time goes on, but there will always be a need for fairly complex pricing structure in offering this kind of service. The problem is just not as easy as it sounds.

Now let's talk about what services are commercially available. The Time-Sharing System Scorecard published by Computer Research Corporation, 429 Watertown Street, Newton, Massachusetts, lists many of the academically-oriented time-shared systems in operation today plus thirteen different commercial enterprises already in operation, including CEIR, IBM, GE and KEYDATA. These and most of the others are concentrated on the East Coast and West Coast, with Comshare in Ann Arbor being practically the only one in the Midwest. Except for KEYDATA and VIP Systems (which is concerned with text editing) most of these are very similar in concept and supply the kind of facility you have seen demonstrated - they are user-programmed facilities employing languages such as Dartmouth-GE Basic, Rand Corporation JOSS, IBM QUICKTRAN or others. While they are similar to what you have seen, most omit entirely anything equivalent to Ford ability to connect to the high-powered Philco 212. In other words, the front end of the Ford system, working in the GE235 is quite typical of what you can buy from any one of a dozen organizations. Some of them have better file handling capabilities than others, and I suggest that you find out which those are. Otherwise you may have to play around with paper tapes, storing and keeping track of your own programs, and you will not have as attractive a time-shared facility as you otherwise might.

As a closing note, I would like to spend a few minutes dreaming with you about where this whole thing is going. Remote-access computing was first demonstrated as a physical concept not in 1965 or even in 1945, but in 1939, from Dartmouth College to a Bell Laboratories computer in New York. So remote consoles, as such, have been around for a long time. Time-sharing in the sense of multiple-access storage and processing devices, conceptually originated with such facilities as the American Airline Reservoirs and the John Plain Company mail-order processing and inventory system back in the early 1950's, all of them built around magnetic drum systems. The SAGE air defense system was probably the first highly-flexible, heavily time-shared operation.

Thus, the two technologies of remote communication and time-sharing have been around for a long time, but the development of hardware and software techniques to facilitate the kind of efficient use of multiple-access computing going on now was originated recently and is still a very new subject. Even many of the organizations listed

on the previously-mentioned "Scorecard" are on the air in only a limited sense. That is, virtually all of them are having problems with one thing or another, and many of them are just barely getting going.

Time-sharing as a commercial enterprise is still a novel field. But it is going to move very fast. Western Union, GE, IBM, ITT, CDC and others are already active in a number of things. With so many large organizations moving in that direction the FCC has become interested, which will further complicate life I suspect. It is going to be an interesting area.

The role of the FCC is an interesting question because there are no clear-cut boundaries. What KEYDATA is doing involves communications line but not real communication; it merely brings the computing capability to the local subscriber. The Westinghouse time-sharing facility on the other hand is primarily a message communication system with some processing thrown in. If you look at what KEYDATA might be doing in a few years hence you can readily envision someone saying, "Here is Comshare in Ann Arbor. They have a library facility that they can make available. Why not let KEYDATA subscribers request access to it, so that the KEYDATA computer communicates with a Comshare computer, gets the answer and feeds it back to the person who asked for it?"

One can imagine a wide variety of interconnecting communications links and this will certainly raise some interesting questions as communications and data processing move together. I personally believe they are going to move relatively rapid. Technically they could move so fast that a complete intermarriage of all media of communication into a generalized one is possible within at most a decade. Social, political and economic problems may well slow this down, but there appear to be no serious technical difficulties.

A wide variety of remote consoles is also in prospect. KEYDATA uses Teletypewriters because they are there, so to speak. The phone company owns, installs and maintains them; they are fairly reliable and inexpensive but they are a bit clumsy and slow. You see a sparkling new graphic display device and think perhaps you would rather have one of these. For many applications, anyone would. Hard copy is needed for some things and useful for many but not for everything. Cathode-ray tube displays of alpha-numeric or graphic data are indeed useful; they just happen to be a bit expensive for an individual remote installation, since they require a fair degree of processing

locally. Audible response is likely to be an increasingly important part of the man-machine system. The "Touchtone" telephone will become one of the important terminal devices in the relatively near future. It already is in use on the New York Stock Exchange. You can inquire about stock prices by keying on a Touchtone telephone and listening to the answer through the telephone handset.

It is going to take some time for these things to evolve. It is going to take time for file storage costs to come down, and for file organization to develop into a science that permits handling very large files inexpensively. The display devices will become compact and inexpensive using as a processor a tiny computer, probably more powerful than the present small computers but even less expensive. Things will get more sophisticated and ultimately communications, data processing and information retrieval will merge.

Before too many years have elapsed, you will be able to carry with you a pocket communicator containing audio and visual outputs, easily operated input, a fair amount of processing, and a wireless connection through which you may communicate with your friends or with echelons of computing and file storage facilities, if you wish. If you want to search the Library of Congress you will send a request that will go through one device after another until it finally reaches the data that you want. If you want something very simple computed for you, it will be done by the tiny computer that forms part of your pocket communicator.

I have with me a mock-up of such a device. Unfortunately, it is not working yet because there is no network behind it, but as you see, it can be held in one hand like a book and features a small luminescent screen that displays alpha-numeric and graphic information. It can easily be operated by a child. By pressing this button, you display a list of the classes of data and services you might want. Push the button up and the list moves up. When what you want gets to the top, push the button to the right and presto, another level of detail appears. Again, you move up, and over, and up and until you have found what you want. Perhaps you move over on "periodicals," then up to "news magazines." Push over to get a list of news magazines, push the button up until you get "Time," then over to get its Table of Contents, up to where you want to start. If you want to see the text, merely push over one more level of detail and read.

If you just want to read the editorial content, it will cost you money because your service meter will be running while you do. If, on the other hand, you are willing to look at the ads while you are reading your magazine, the companies whose ads you are looking at will be charged for the service to you in proportion to the time you

spend on their ad. (If you show much interest and are in the right income bracket, you may even get on their "mailing" list.) If you like what you see in the ad and want to buy it, push the "buy" button and the product is yours.

As you can see, my optimism for the rapid development of the computer utility is quite great. But in closing, I feel I should comment on the one thing that I think is going to be slow in being implemented: the ability to communicate complex data into a computer, both in terms of mechanics and of semantics. I put a full, though compact, keyboard on my mock-up communicator because I think keyboards are going to be around longer than many other people believe. You and I may have some trouble with keyboards because we are not typists. Our children will not, because they are already starting to use keyboards even in the elementary grades and this seems likely to grow. As you probably know, remote-access computing is already available routinely in a number of high schools in my home area of Boston and elsewhere, and it is already here in the third and fourth grades in Stanford and Palo Alto. Keyboards will very likely be second nature to all children born after about 1970. Thus, one of the things that often slows down the man using remote-access computing today, the fact that keying is simply not his natural way of communicating, will gradually disappear.

Many critics of today's machine-aided design experiments using time-shared facilities make nasty comments about the available languages. I think one of the things they overlook is that some problems are pretty hard to state in any language simply because they are tough problems. It is often not a limitation imposed by the language per se. Anybody who attempts to express a complex problem in clear English, for example, without using the language of mathematics, chemistry, mechanics or whatever, is retrogressing. We have evolved these special languages so we can express complex things easily, even though they still leave much to be desired. What most computer language researchers are trying to do, then, is to let the computer understand the language of mechanics, chemistry and so on - not natural English, which is probably the worse possible language to try to teach a computer. And with that thought, I leave you.

DISCUSSION

Question: In KEYDATA's accounts receivable service, what percent of your subscribers are using identically the same programs?

Mr. Adams: Our initial subscribers were recruited during a period of experimentation a year or so ago when we took all comers. They represent a miscellaneous sampling and were sold on the idea that we would give them whatever it was they needed. Each of them wanted something different than the others, with the exception of one group of three different distributors who handle the same line of goods and use the same physical program. While situations such as the latter may increasingly arise, we really do not expect many subscribers to use identical programs. One of them wants to do discounting one way, another in another way; a third does it one way for one kind of customer and another way for another kind of customer, and so forth. We try instead to offer a variety of standard options from which to choose. As long as subscribers pick options that are already available, we can piece them together just as the auto-makers piece together options to make a very wide variety of different cars on a production line basis. When subscribers want to do things that really differ in some fundamental way from our standards, it will cost them money. We find that when you present businessmen with a fair statement of how much it is going to cost them to do things differently, they tend to conform.

Thank you very much.

Norman Scott:

Thank you very much, Mr. Adams. I think the next item is for us all to adjourn for the noon hour.

AFTERNOON SESSION

Norman Scott:

Our first speaker this afternoon is Mr. Richard F. Lewis from the Clark Equipment Company in Buchanan, Michigan, who has chosen the very stimulating and interesting title "Emancipation of Engineering from Finance Oriented Computers." Mr. Lewis is a Mechanical Engineer by training. He received his bachelor's degree in 1958, and entered the computer field as a scientific and engineering programmer with the farm equipment division of the International Harvester Company, working primarily in the structural design area. Later on he transferred to business applications in their material control and warranty and field failure analysis areas. He joined the Clark Equipment Company in February, 1963, as corporate coordinator of engineering computing. He is presently engaged in the development of engineering specifications for a management information system. I think he will have some very interesting things to tell us this afternoon.

EMANCIPATION OF ENGINEERING FROM
FINANCE-ORIENTED COMPUTERS

Richard F. Lewis

Advanced Systems Analyst
Clark Equipment Company

EMANCIPATION OF ENGINEERING FROM
FINANCE-ORIENTED COMPUTERS

Mr. Adams made reference to the fact that I may have an axe to grind with the accounting people. This is not so. As a matter of fact, formerly, our data processing organization revealed that I was a function of accounting, a member of a financially oriented computer system, and indirectly reporting to the controller of the corporation. The point that I was trying to convey, and I could find no better words to describe it other than "emancipation", is the frustrations that I, as an engineering representative in the area of computers, was burdened with over the past eight years or so. As we all know, computers were originally introduced to business for the purpose of payroll preparations and other financial data reduction. Consequently they fell under the responsibility of the controller, and that is where they remain, unless you are a large enough organization like the Ford Motor Company whereby Engineering can have an elaborate system of their own, such as was described for us earlier.

I can remember the "good old days" - I say it with quotes - and you who have been engineering programmers at the mercy of the financial computer people know what I mean. We sat down with the boss and told him that we were top rated programmers. The boss said, "Fine, we have a little problem, and these are the parameters. What do you think about it?" The immediate response was, "Well, I think I can get it done in about four hours." So we sat down and started coding. We forgot that after it is coded on a sheet of paper you turn it in to the key punch section. Now the keypunch people are very proud of the fact that they can provide you with a 24-hour turn-around service. The next day operations people at data processing, who are happy about the fact that they can provide a 24-hour turn-around service providing it is not a payroll day, end of the month, special job on the ledger, or a rerun, accept your cards. However, it is one of those days, but there is the opportunity to come in at midnight or Sunday morning at two. Well, anyhow, we are now into the second day and possibly on the fourth day we get the program deck in to the computer and it comes back with a typewriter log which says, "No END card." We have not even gone into execution yet and we are now in our fifth day, on this "four-hour" job. You finally correct the program deck and possibly on the fifth or sixth day you get into execution. It happens to be a recursive type of problem. You are looking for an optimization, but this "business" computer operator sits there and does not see the printer going at a full 600 lines a minute. He figures you are in a loop, so he takes a memory dump and whatever else there is, and returns it to you. Maybe by the seventh or eighth day you finally convince them that this is the way

it goes. This is that "four-hour" job. Then timesharing came along and when we obtained the services at Clark Equipment Company, all I could think about was "I wouldn't have to walk into the boss's office with a red face again."

I will give you a little background on Clark Equipment Company. I do not know how many of you people are familiar with the organization. We are a multi-division, multi-plant organization and one of our goals is not to force centralization, as was indicated to be a trend by one of the previous speakers. We do function in a decentralized mode; however, I think that it is necessary to centralize the store of information, and the dissemination of this information, primarily for control purposes. Clark has quite a few operations, mostly here in Michigan. We have made some new acquisitions. Recently we have acquired the Hancock Corporation in Texas, which is a scraper manufacturer. I also noticed in the paper last week that we acquired a foundry. Our sales for 1965 were about 392 million dollars. We manufacture fork-lift trucks, straddle carriers, also electronically-controlled material storage and retrieval systems, which are our latest. We also manufacture automotive components such as transmissions, torque converters, axles, and axle housings. We are also in the commercial refrigeration business, and earth-moving equipment, so we are quite diversified. Clark Equipment Company some twenty-five years ago entered into the data processing business using EAM equipment. Our present computer configurations consist of twelve medium sized computers. These are finance-manufacturing oriented, finance-manufacturing controlled computers. Engineering users do get consideration, however only when we protest loud enough to make our points. Part of this problem has been our own. When you talk to the data processing people in regard to "production schedule updating" they can visualize what is happening, as well as understanding the significance and impact of the processing of this information on a business. But when you give them an engineering type problem they really do not know what you are trying to accomplish. They are unaware that there are intangible as well as tangible benefits to be derived when you are able to computerize some types of engineering problem-solving. Rather than assume that they wouldn't understand what we're trying to accomplish, we might have been better off by processing them with more insight.

What we are attempting to do at Clark, and we hope to be on the air by the first of the year, is to hook up four remote computers to the computer operation at our corporate headquarters in Buchanan; these are for our four major operating divisions. This is via telephone communication line. The purpose of this is to be able to input the large volume of operating data generated at the divisional level, process and update the files located at the corporate center, and do

whatever reporting that has to be done. Our efforts, in what we call Phase One, are primarily for material planning and control purposes. Three years ago we were extremely ambitious and thought we were going to do a complete job. As we got into the development and design phases of the project, we realized the complexity and task required to accomplish such an undertaking, so we decided to reduce our scope.

Our computers, with the exception of one or two, are furnished by the General Electric Company. The corporate central site in Buchanan consists of a G.E. 235 and a G.E. 225, including nine disk files and one Datanet-30 communication message switching computer. Our remote satellite computers are G.E. 215's and G.E. 225's. About nine-tenths of what might be a time sharing system offered by the General Electric Company is on site. We are lacking an auxiliary Arithmetic Unit, and the necessary software. The first thing that comes to mind is, "Why don't we do our own timesharing?"

To be able to answer the question, I'll need to give you a little background on our use of the Datanet-30 communication switching computer. At the present time it is being used primarily for message switching between our Dealer network and our manufacturing facilities. Each of our dealers has a teletype, and in the past a message being routed on our "torn tape" system would require it being received in Buchanan on a punched paper tape. This message would then be stored in racks, and later physically mounted on another Teletype unit for transmission to its destination. This is now being handled automatically by the Datanet-30. When our Real-time On-line Material Planning and Control system becomes operational, all data transmission will also be handled by the Datanet-30. They tell me that we have about 9000 words of program in a 8000 word core at this time. We do not have the memory capacity to even consider its use for timesharing at this time.

As you can see, timesharing is not only for the small users who only have a couple of engineers and no computing facilities available, or for the extremely large user like Ford Motor Company, but also for an inbetween user like ourselves. We cannot afford our own timesharing system at this time. They tell me that General Electric will not lease a 265 timesharing system. It has to be purchased. I think the figure quoted me was about \$725,000, and our engineering management is not convinced that we need computers to the extent that they are willing to allow us to spend that kind of money. Although I think the day is coming.

We were introduced to timesharing services as a utility, which was being offered from Chicago. February 1966 happened to be the month of one of our engineering council meetings, and at that time it was

brought to the President's attention that engineers were not obtaining the computer services they needed. Coincidentally, we arranged to demonstrate timesharing to this group. They were impressed by the fact that we were able to dial into computers in Phoenix and New York. At the time that we conducted the demonstration service was not available in Chicago and we made arrangements in Phoenix to obtain our timesharing services. We didn't get into Phoenix because of a busy signal, but we knew what computer demonstrations were like, so we made alternate plans with New York. I think this was the first time that the mysterious covering, which has been hovering over computers, had been removed for some of the chief engineers who attended that meeting.

I think that the most impressive thing to these chief engineers was the fact that they would not have to be confronted with the complexities of a giant computer. They did not have to fool around with switches and dials. They could sit down and simply communicate with the computer. When an individual sits down at a teletype terminal for the first time, knowing that he is going to be communicating with a computer, and he goes into the salutation sequence, and that final greeting on that salutation says, "ready", I think he is committed to timesharing. He realizes the ease with which he can become a computer user.

The results of that demonstration prompted a decision that a representative from each division would formulate a committee to decide whether or not we want to get involved in timesharing. Were our own computer services that bad, or weren't they? We decided that there were certain favorable conditions which would prompt us to at least try timesharing for a brief period of time. This is the beauty of it. You really don't have to sell it as a two-year plan, or five-year plan. All you need is 30 or 60-days to introduce the system. We purchase our timesharing services from Chicago and our contract can be terminated upon 30 days' written notice. Bell Telephone Company will disconnect the terminal and stop charging you fees the day that you call them. So you don't have much of an investment. If you are willing to gamble those few dollars which it will take to satisfy the basic charges of the contract, I think that this would be the extent of your expenditure. You are not looking at a million dollar contract or large budget appropriation. We had the units installed on the basis of a 90-day trial period.

One consideration, which was favorable, is the fact that Clark Equipment Company has an existing Centrex telephone system. I do see figures quoted sometimes where the telephone toll rate is greater than

the actual charges of the computer itself. When you have accommodations as we have, with ten direct lines from Buchanan into Chicago, operating through a Centrex telephone system, our phone bill is virtually nothing. We are paying for the lines anyhow.

I think that our prior Algebraic language experience had brought to light the ease with which a language like BASIC can be learned, understood, and used. There wasn't anyone that we had introduced the BASIC language to that wasn't able to use it. I do not recall that anyone said, - "Well, it is beyond me." It is really a simple language. I am sorry we did not have TUTOR I through XX as was described in the Ford presentation. The other thing is that if you are planning to introduce timesharing, you are going to require a training period. FORTRAN classes are usually a week or so in duration, which is about the minimum. This represents a tremendous investment in training costs. Our training session on the BASIC language lasts four hours. Everyone that attended one of our BASIC training sessions has developed a program of his own.

The location of the terminal is important, and was brought to my attention very vividly six years ago when I visited the Buick Division of G.M. I talked to Bob Loudon, who was the manager of technical computing out there at the time, and he said he would like to draw diagonals across the engineering department. Where those lines intersect, would be the location of the IBM 705, which is what they were using as an engineering machine at the time. You have to make it convenient to engineers. You cannot have it across the street, or they are going to be reluctant to walk across the street and use it. It should always be in their presence, and by mere association with the equipment, they will become a little more intimate with it.

We have talked about the cost - the minimal amount of cost that was involved. The minimum subscription for the General Electric System is a base charge of \$350 a month, which includes 25 hours of terminal time, which is the connecting time, and 7200 seconds of CPU time, including 40 storage units for your programs. This \$350 a month equates to half of a man, including overhead, and is very worthwhile. This low cost appealed to us.

We had other plans for the teletype machines beside their use as timesharing terminals. Soon after the first of this year, when our Real-time On-line Material Planning and Control system gets on the air, we hope to be able to make engineering inquiries into the corporate master part record file. This should allow the engineer to get the latest costs on parts, usage, and so on. When an engineering change is necessary, the engineer will at least have some current and relevant facts to back up that change. This terminal, via the Datanet-30, should

be able to provide answers fairly readily. This is no idle dream; this is something we hope to be able to do soon after the first of the year.

The other area of use for these terminals is to implement a concept that we have been working on for some time. This is the automatic generation of Bills of Material from sales orders. IBM has a similar package they call AIDS. What our system amounts to is to take the customer requirements appearing on a sales order, process them through a file of information organized in tabular form, and generate that unique Bill of Material which satisfies the customer requirements.

There was some mention by Mr. Missler about corporate policing of the terminals. This is something we thought about. Will these time-sharing terminals, which advocate open shop policies, lead to misuse of the system, of computer time, etc.? As was Ford's decision, we decided that we are not going to meddle. As a matter of fact, from our corporate terminal we cannot access any of the divisional user numbers. We do not want to look at what they have in the library. Our corporate office gets involved in divisional timesharing programs only when our assistance is requested. If, at anytime a user desires to determine the angles of triangles, this is alright, because I think it is part of the learning process and I think that it is necessary. However, I did make an unofficial inquiry into the divisional libraries to see what kind of problems they had been working on. Were they real problems, and not just academic exercises? The libraries contained real problems which they were able to get on the machine, problems related to their areas of endeavor, and I was amazed and pleased.

I don't know how many of you are operating in the environment of labor bargaining unit in your office, but we have one division that has a unionized office staff in the engineering department. It was brought to my attention by one of our engineering managers that this may present a problem and I should be prepared to get involved in arbitration as soon as it is installed. One of the things we didn't want to happen was to have some clerk sit down at the teletype unit as an operator; I think a lot would be lost. The engineer who has a problem and programs it himself is able to exercise his imagination a little more. If at this one division, a trained operator were placed at the terminal we would run into the same problems we had in the past. As it worked out to the amazement of our Industrial Relations Department, we had no complaint from that particular bargaining unit.

Our first two months of operation were discouraging. We had more than our share of problems. I called Whirlpool, a neighbor of ours in Benton Harbor, to ask if they were having the same problems we were. They indicated they were, but not to the degree we experienced. These

were nuisance types of problems and what it boiled down to was that our telephone lines were only of voice grade. They were not in balance and consequently we were getting some feedback which resulted in garble.

Our terminals at Clark are primarily Teletype 33 ASR's. The four basic types available are the 33 ASR, 35 ASR, 33 KSR, and 35 KSR. The difference between a KSR and an ASR is that the KSR does not have the automatic-send capabilities, which we felt were necessary. We felt that there are those engineering problems which will require an abundance of input data and, consequently, we would want to punch them off-line for economical reasons. We did have our problems with the terminals; but Bell Telephone finally got those adjusted. This I understand is an inherent characteristic of the 33 ASR. When you dial the computer, the computer responds to the terminal unit with a "Who are you?" signal. This trips an answer-back drum, and what happens then is they try to match up the code contained in the answer-back drum with your subscription identification, and what is typed into the terminal as the user number. This validates you as a user. We said that we had a dual purpose for these terminals. The primary function is to get into the time-sharing system. The secondary reason is to access our own Datanet-30 system. Our Datanet-30 communication switching system was existing before timesharing became available to us, and we had our own "answer-back" configuration to satisfy also. We were able to make up an answer-back drum to satisfy the requirements of the timesharing service in Chicago as well as our own but this turned out to be one of our problems. We were getting invalid terminal messages. So we went ahead and put on answer-back drums which satisfy the timesharing service configuration only. This had resolved a considerable amount of our problems, but prohibits the use of our own Datanet-30.

Presently we are enjoying excellent service. The timesharing service offered by G.E. in Chicago is operating at least six days a week and usually about 20 hours out of the 24. It does have its peak loads but I cannot recall ever getting a busy signal from the computer itself. We get more busy signals from our own telephone lines going into Chicago than we do from the computer. We did have some problems with the timesharing system, but thank goodness they had a backup tape for all programs, which they restored for us after our programs were wiped out of storage. We had 40 programs dangling out there on the disk when this system malfunction occurred in Chicago. Why it chose the Clark Equipment Company area of storage, I do not know, but it obliterated us from their files. As it were, the worse that happened to us at the time was that whatever was applied to modify existing programs on that particular day was not in the system when they re-loaded us. I had visions of special engineering council meetings and having to pull teletype terminals out myself.

I know that we are not the Chicago system's largest user; however, we are one of their five largest users. They do have a user who last month used about \$30,000.00 worth of timesharing. Our use is about \$1,500.00 to \$2,000.00 a month of system-service-charges. This accounts for about 100 to 150 hours of terminal time, with about an hour to an hour and a half of CPU-time and about 100 units of storage. On the average, we are using about 75 hours of terminal time for every hour of central-processor time, which we thought was a little inefficient. The reason I became concerned with this is that the basic subscription ratio of allowable terminal time to CPU-time is about $12\frac{1}{2}$ to 1. If we compare 75 to 1, to $12\frac{1}{2}$ to 1, you might say that there must be some existing inefficiency. After doing some research, we found that the inefficiencies were not as much ours, as they were the billing structure inadequacies of the General Electric timesharing service. I do not think anyone in the timesharing system is getting a $12\frac{1}{2}$ to 1 ratio. The ratio increases to about 25 to 1 for that user who has sophisticated programs, whereby the program uses the central processor for an extended period of time. The longest central processor running time that I am aware of is about 40 seconds for any one of our problems. We are primarily in the one- to ten-second range. We spend more time communicating than we spend running.

Timesharing, as it is known now, is not timesharing in its pure sense. It is pseudo timesharing. We do not have the file capacities needed to satisfy some of our other engineering problems in the administrative areas. We do not have the capabilities in a system like the General Electric timesharing system to run a CPM network of a large size. We are still dependent upon these financially-oriented control computers for this purpose. I think we may have them worried because our usage is dropping. The administrative type of activity like engineering labor reporting, labor accounting, and budget reporting, which requires storage of historical information, cannot be economically accomplished on timesharing.

As we go from research to design, we find that more and more time is being spent in information retrieval; digging back into historical information. I am sure that there is a lot of this information that can be captured and stored in a computer, or in a computer environment, whereby a remote terminal would make it real handy for retrieval purposes.

As more engineers became involved with timesharing, we found that we were trying to solve problems which had not been computerized in the past, such as specifications for customer requirements. As an example, to prepare a bid for a specific piece of machinery we'll need to know what combination of torque-converter engine and drive-line which will be needed to satisfy the specifications of the contract.

Well, in the past, somebody made a guess at it and came out with a price, and we found that a majority of the time we were wrong. We found that the guess did not satisfy the requirements of the customer. With the availability of a timesharing terminal right near the engineer, he is able to run a performance simulation and determine what would be the combination of drive line components which would satisfy the requirements of the contract. In one case we found that a bid had been proposed which was evaluated manually and proved to be infeasible on the Performance Simulation program. We did not have an engine at the time plus a torque-convector that would satisfy the requirements.

This seminar or symposium is oriented towards the impact of timesharing on engineering. The impact is not any more significant than that which engineers have found to be the impact with any unlimited access computer. Analytical analysis and computer reduction of data is the big payoff. Instead of taking a scraper blade out, pouring a concrete pile, then taking this prototype and ramming it into the concrete pile to see if it falls apart, this is too costly. We are seeing less of this type of eyeball design being produced now. Our engineers are taking a more analytical approach to problem solution. They are providing themselves with more of the right kind of information, at the right time, for the purpose of optimization.

As an example, let us consider a turbine blade on a torque convector. An engineer will say, "I only have this much time in my schedule to get the job done; therefore, I will look at only that number of configurations of blade angles, etc. which I can evaluate within a short period of time." With access to a computer, he is generating more design information. He is able to take a look at and optimize, from a selection of 50 to 100 blades, in less time resulting in a less costly design. Intangible benefits are pretty hard to define, but they do exist.

I would like to see a simple language, such as BASIC, available to the every day design engineer because this provides him with an additional tool. I would hate to see them become computer specialists. When a creative engineer becomes computer oriented, there is a tendency for him to spend more and more time with the computer and consequently he learns more sophisticated languages. He winds up doing more computer work than design work. We lose him as a creative engineer in our design areas. We need a simple tool like an extension of the slide rule which would allow him to do his job quicker and easier. A timesharing terminal and a simple computer language is this extension. We find many of our engineers who previously have been afraid of computers are not afraid to sit down and work with this one. Due to accessibility, timesharing is an engineer's dream. It allows programmers to do on-line "debugging."

When a programmer starts doing on-line debugging he gets the job done. As he gets more sophisticated and starts using the in-house computer he doesn't get this "hands on" participation because of the scheduling problems it presents.

Timesharing is a computer philosophy which is considered in other functional areas besides the engineering department. Clark Equipment Company has plans to install, by the middle of 1968, one of the new third generation computers, which is timesharing oriented. We are looking forward to this being a corporate philosophy. It is our intention that all functional areas of Clark Equipment Company will be timesharing by 1968. This will involve a variety of terminals. We will have some at the speed of the teletype terminals and others, which have high input-output capacities, like the Univac DCT 2000.

Recently one of the people from accounting came up and asked if he could use our timesharing terminal. He wanted to do some regression analysis on cost figures, and wanted it done today. Our operations research people are finding it advantageous also. They, too, were suffering from lack of computer accessibility. I can recall a couple of projects recently which were oriented towards forecasting, economic order quantities analysis, material ordering etc. They were business data processing problems. The mathematical model and its individual parameters were tested with the use of a timesharing terminal and with limited data. They were modified and retested until the ideal models were developed. They have since become that part of the larger production program, which went into the final system design. They did not have to go through the seven day turn-around time that was mentioned earlier. They, too, were able to take advantage of accessible computer services. Gentlemen, I want to thank you for your attention, I do not know what else I can tell you about the Clark system. I will entertain any question that anybody might have.

DISCUSSION

Question: You mentioned that one of the uses of your computer terminal was the automatic generation of Bills of Materials by the use of a technique similar to IBM AIDS. Do you do any cost estimating along in that package?

Answer: No, we are not at this time. We do have it in our future plans. Our primary objective now is to see whether or not this technique is applicable to our type of business, that of automatic generation of Bills of Material. If we can accomplish this, I think that we have a whole spectrum of areas we can go into. Thank you.

Norman Scott:

Thank you very much, Dick.

Our next speaker is Professor Bertram Herzog. I feel it is appropriate to have at least one speaker from The University of Michigan to tell you about some of our own developments in this area. This spot will be filled by Professor Herzog, who is Associate Professor in Industrial Engineering here at the University.

Professor Herzog is a graduate of Case Institute of Technology and received his Ph.D. in Engineering Mechanics at The University of Michigan. He served here as an Assistant Professor of Engineering Mechanics and then joined the Ford Motor Company for two and a half years where he was Manager of Advanced Computer Systems Planning. He rejoined us in 1965. Bert has been heavily involved in computer graphics and on-line computer systems, and I know he has some very interesting things to tell us.

Professor Herzog.

REMOTE TERMINAL EQUIPMENT FOR
TOMORROW'S DESIGN ENGINEER

Bertram Herzog

Associate Professor of Industrial Engineering
The University of Michigan

REMOTE TERMINAL EQUIPMENT FOR TOMORROW'S
DESIGN ENGINEER

Thank you! I made the mistake of trying to adjust several mechanical things before the last talk, but it started before I could finish adjusting; so my notes are here, and I have not had a chance to review them.

I would like to take the posture this afternoon of being an engineer, which is how I started out in life, and I guess I really continued that way. I was seduced into the computing business in a rather peculiar manner, partly by my dissertation, and partly by the Ford Motor Company. I am concerned with the use of computers by engineers, and the topic of my talk this afternoon has something in it about the future. Norman Scott dug up this thing about advanced computer systems planning, but I feel somewhat bankrupt in this area. Things change so rapidly here that even since I agreed to give the title to this talk things have changed, and I imagine some people in the audience are better qualified than I to give this talk, but they were able to avoid doing so. I am left not only in the position of being the last speaker, but now that the deer hunting season is coming up, I'm worried about finishing on time. Deer hunters are not only vociferous, they're also armed.

The concern I have with computing and engineering, and we are talking, I hope, in the context of engineering, is that in past years to use a computer we had (a) to become computer experts, unless we could afford to hire computer experts (several people have addressed this question) and (b) to learn computer programming. In the process, a good many engineers were seduced, probably to the benefit of the computing business, into the computing business itself, which of course is an engineering business, and it ought to be approached that way sometimes. The on-line terminals that you've seen here and the demonstrations you've seen today have, in some respects, emancipated the engineer from having to do all this learning. Although all the speakers have admitted that sooner or later there will be some problems in terms of how big a problem, how accurate a solution, and so on. All this simply says is that eventually we will come to the ultimate position where engineers will interact with computer experts. But the big advantage, I think, has been demonstrated today. It is that many engineers have overcome the impedance of using computers--their reluctance to use them--by the presence of terminals like the one you've just seen. I would like to see this situation improve even more. I do not think the medium of expression of the teletype is the ultimate and best medium of expression.

This leads me somewhat into the question of computer graphics or graphics in general. While I was still at the Ford Motor Company, I was given the opportunity to speak to some educators, before rejoining them so to speak, and I shook up many of them by saying "I look forward to the time when we do not have to teach FORTRAN or some other programming." It was necessary then. Had we not spent those hours initially we would not be where we are today. That is part of the pioneering and readjustment problem. Looking ahead to the future, I am very jealous of these hours. These terminals make it easier for me to introduce, in my course, certain ideas without having the student go through a whole palaver of computer programming. The same problem I fear is evermore important, in terms of dollars and cents, in industry. I cannot afford to train a large group of engineers in FORTRAN programming so that they may use a fast-processing computer. So I want to go one step further now. This leads somewhat to the future, and it leads in some sense to the past; I want to go into computer graphics. I want to communicate graphically, which even removes this sort of device (teletype) in some applications. I would not go so far as to say what the mixture of these things might be. The things I do want to see happen is ease of use of these devices for communicating with computers and a flexibility of locations. As a matter of fact, with respect to flexibility of location, I would like to interrupt my talk for a moment, go outside, and bring in some equipment.

(Demonstration of Dataport, essentially a teletypewriter in a suitcase, which requires only an ordinary telephone to give the user on-line access to a large, central computer.)

I would like to emphasize, in part, the ancient nature of computer graphics--and there are several of the pioneers of this field sitting in the room here. With thanks and appreciation to them, I will go through a selected number of slides. (This is an opportunity that you should really have after lunch, because the lights will go out and you can sleep in a relaxed fashion.)

Figure 1 is one of the most common forms of computer art around the University of Michigan. Some of you may have heard about our possessive attitude toward the MAD language. In the early days, when computing was at a low tide, we used to be greeted with this picture every time we made a goof. The caption, "What, me worry?" is typical of the situation that you faced as a computer programmer. With teletype terminals you do not really worry any more, but this picture was a daily reminder as you waited the days to get your results back. You can get all kinds of weather maps, Christmas greetings, and the like; they are easy to get, and many people use them.

THE MICHIGAN ALGORITHM DECODER
PRESENTS



Figure 1

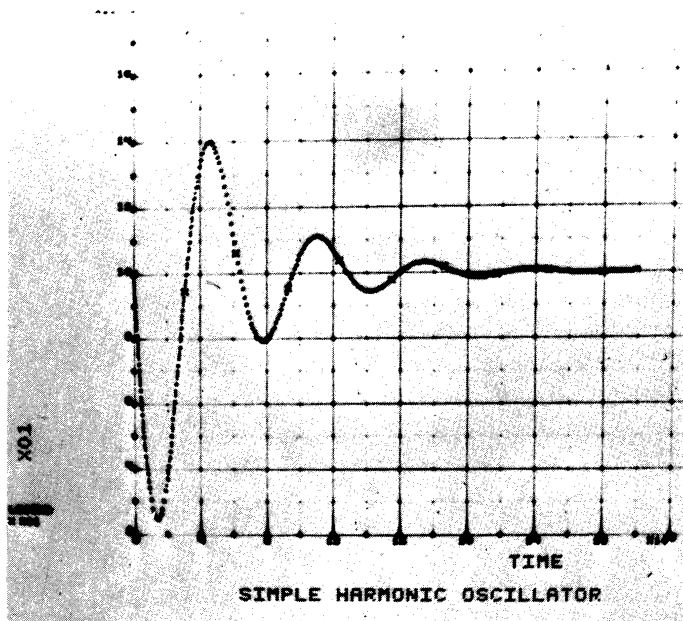
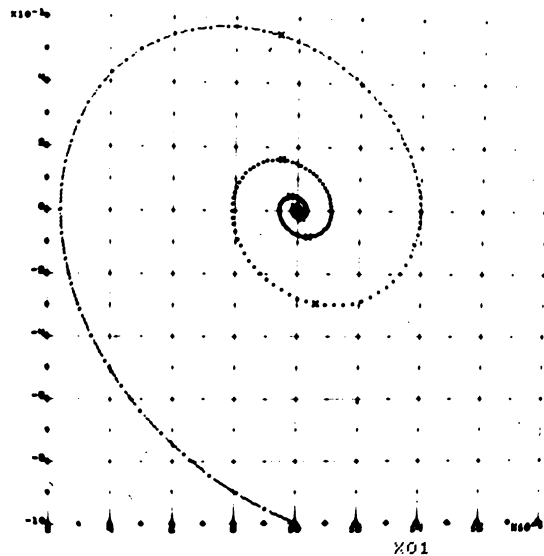


Figure 2

In Figure 2 you see Ford with General Motors routines--part of the trading that went on. This was a rather common form of output for certain programs, but there was great reluctance to use it. It dates back to 1956, and here we are ten years later talking about the future. The future in engineering is that we ought to use some of the things that have been around for quite a while, and this picture shows part of it. There was great difficulty here. The computer operator had to hang on to some film (and you prayed that he had clean film); when the operator was finished, you hoped he would take it off; then you hoped he would take it down to processing; and finally you hoped you would get a return. This sort of thing usually was accompanied by doubling the turn-around time, which is why people went back to the previous MAD picture.



SIMPLE HARMONIC OSCILLATOR

8 59 PM 292 051065

Figure 3

It was obvious that people could use this (Figure 3) for much more useful quantitative results, rather than piles and piles of data. Zerography, I think, as well as other more modern reproduction techniques, is probably going to be the crucial element here to eliminate photographic loop.

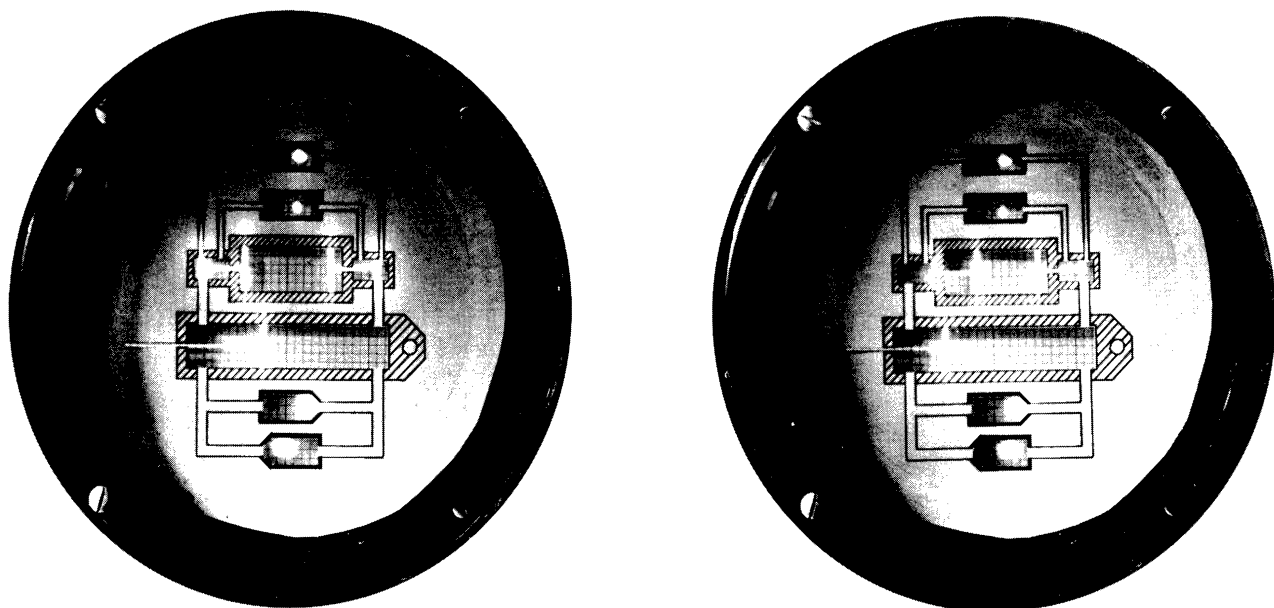


Figure 4

Figure 4 is an interesting historical one. It comes from the Willow Run Laboratories and was done on an analog computer. They were really way ahead. The group out at Willow Run was studying a mechanism for absorbing the shock of a gun being fired. By art work, they masked a cathode-ray tube with the engineer's schematic diagram of the valving system that was being used. I cannot tell you the details of it; I use this picture only to show the application of computer graphics to engineering problems. They had an analog computer whose signals drove little bubbles around, which represented the opening and closing of parts of the valve. To be sure, you are not going to get measurements to the nearest one-thousandth from this picture, but you get an overall system behavior that I think is important.

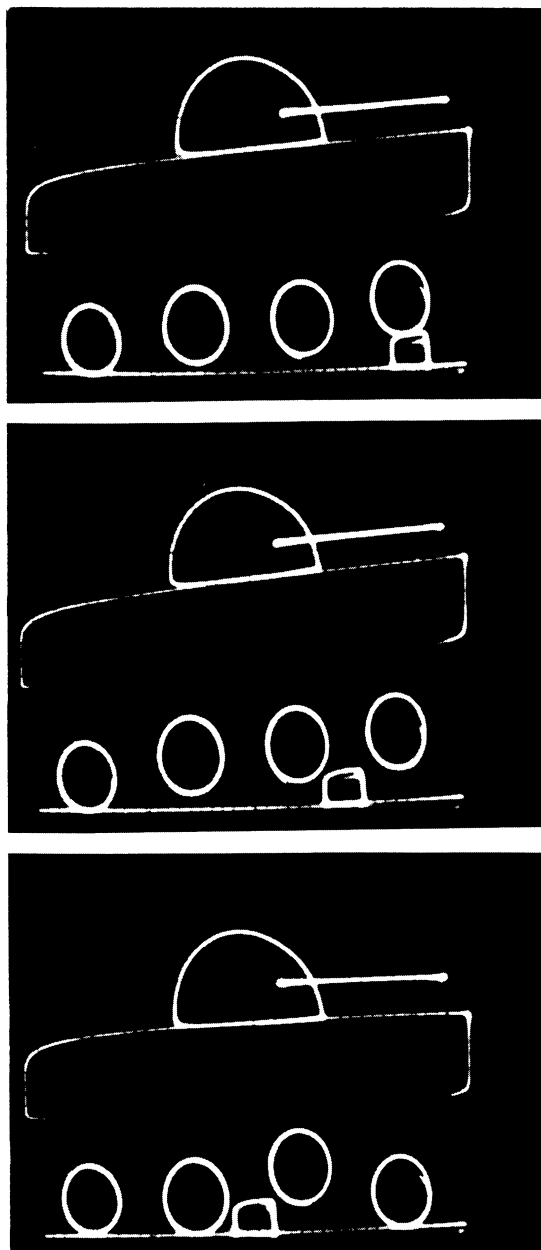


Figure 5

Figure 5 is self-explanatory: it shows the motion of a vehicle going over a bump. Again, it gives quantitative large-scale results, not detailed results. These are the kinds of interactions you would like to have. Unfortunately, the analog computer people were a bit ahead, because they have been a little more real-time than us digital people.

Figure 6 shows a problem that an engineering designer might be interested in. The figure is old, but it serves the purpose of our discussion. The kind of problem that faces many an engineering designer is a mechanism-type problem, typified here by a car window going up and down. In product design there are many, many problems characterized by this figure.

BODY STRUCTURES APPLICATION

WINDOW DROP MECHANISM LAYOUT

PRESENT

- Clearance Test
- Define Objects for Computer
- Process and Plot

FUTURE

- Calculate Clearance and Attempt Redesign if Necessary

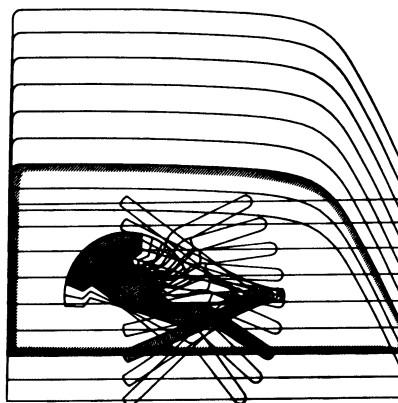


Figure 6

For years and years, the only data processing available to engineers was the two-dimensional sheet of paper and the layout draftsman. He went through many, many gyrations, amounting almost to a multiple exposure of the mechanism's performance, and by this process located the critical regions of interference. For a mechanism that had some speed associated with it, he was further interested in things like accelerations and velocities. You know what a problem it is to get from a displacement diagram down to those levels. The problem, it turns out, is a very simple problem in engineering mechanics; a very simple problem, except when it comes to working it out. It requires a lot of calculation. But every elementary dynamics course in some way or other, in every university, has taught this problem. Except that when the student came to solve it he got stuck; there was too much data processing. One of the features of this problem is that the critical regions are not always clearly defined. It has long been shown that an experienced designer has some value to an organization. He knows where to look for the trouble spots. But, again, he's not infallible. He'd like to know if the assumptions he made about the mechanism going very smoothly over two-thirds of its range of motion are correct, and he'd like to scan that. Well, all right, let's make a layout--every two

degrees. Then he gets into the places where the troubles are, and he starts scrutinizing it in more detail; but he never looks at it in sufficient detail. The spikes in acceleration that you frequently get are the places where you weren't detailed enough in your look at this time exposure of the performance. Now, with a computer, and with interaction with a computer through a graphical medium, you can do this with a computer through a graphical medium, you can do this sort of problem, scan it fairly quickly, and look at the suspicious areas. In fact, I suspect that it's possible to say that you are getting the best out of both parties. Therefore, you will see that some of the things that I will show you, the work of many others, are examples of attempts to use graphics for easier communication. It is obvious here what the problem is, you do not have to write any formula for it. I can write a formula for this, but the optimization routines for this sort of work are very complicated.

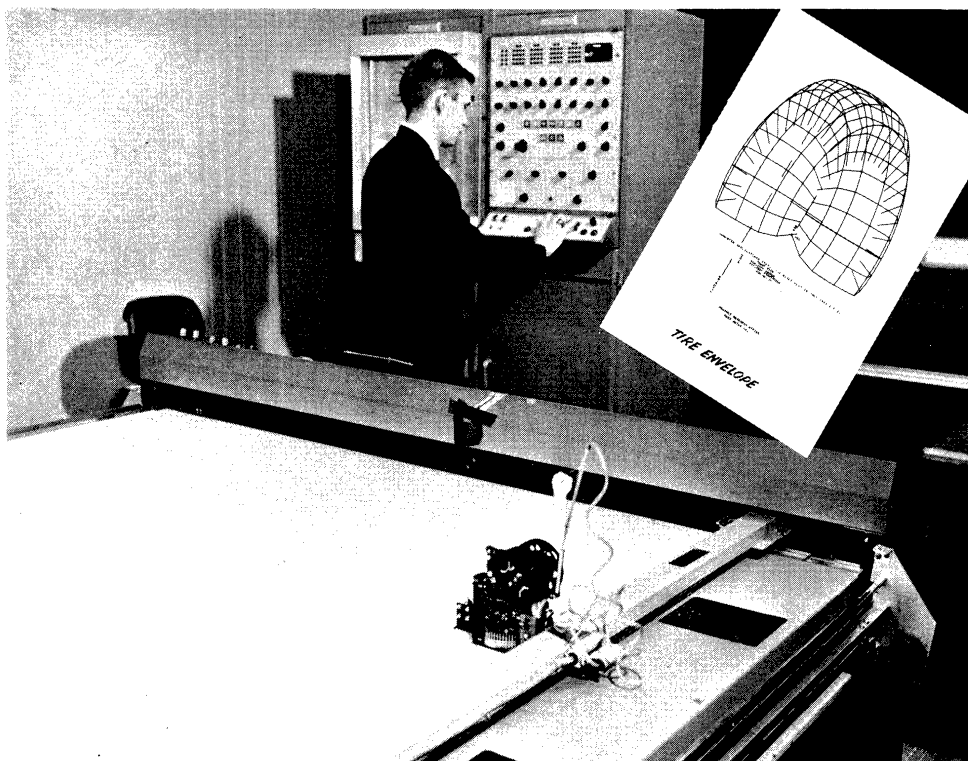


Figure 7

Figure 7 is even more fascinating; I like to think of it as my kindergarten example. Automotive engineers face another problem, that of determining the various positions of the wheel as the car moves along a road so that the surrounding sheet metal does not shred the tires. The solution is to exercise the wheel through all its possible motions. If you know all of them, then you know the places to keep the sheet metal away from. Now, considering a rigid body, you may also worry about tire vibrations and a flexible body. If we didn't educate engineers the way we do--as they grow older and more educated, they become more inhibited--they would go out into a mud pile and fill a box with mud, get that suspension and exercise it like a milling wheel through all of its motions.

When they were finished, they would be left with a mud cake inside the box, which is the surface they are seeking. If they could get rid of their inhibitions, the problem would be solved very quickly. The only trouble is that the suspension hasn't been designed, let alone built, for doing this mechanical exercise. So we have a repetition of the window-drop problem, illustrated earlier, only this time it's a little more complicated. There have been attempts to automate these things, and there have been attempts to have this in an interaction, where the designer can begin to make alterations in the suspension and all the other features.

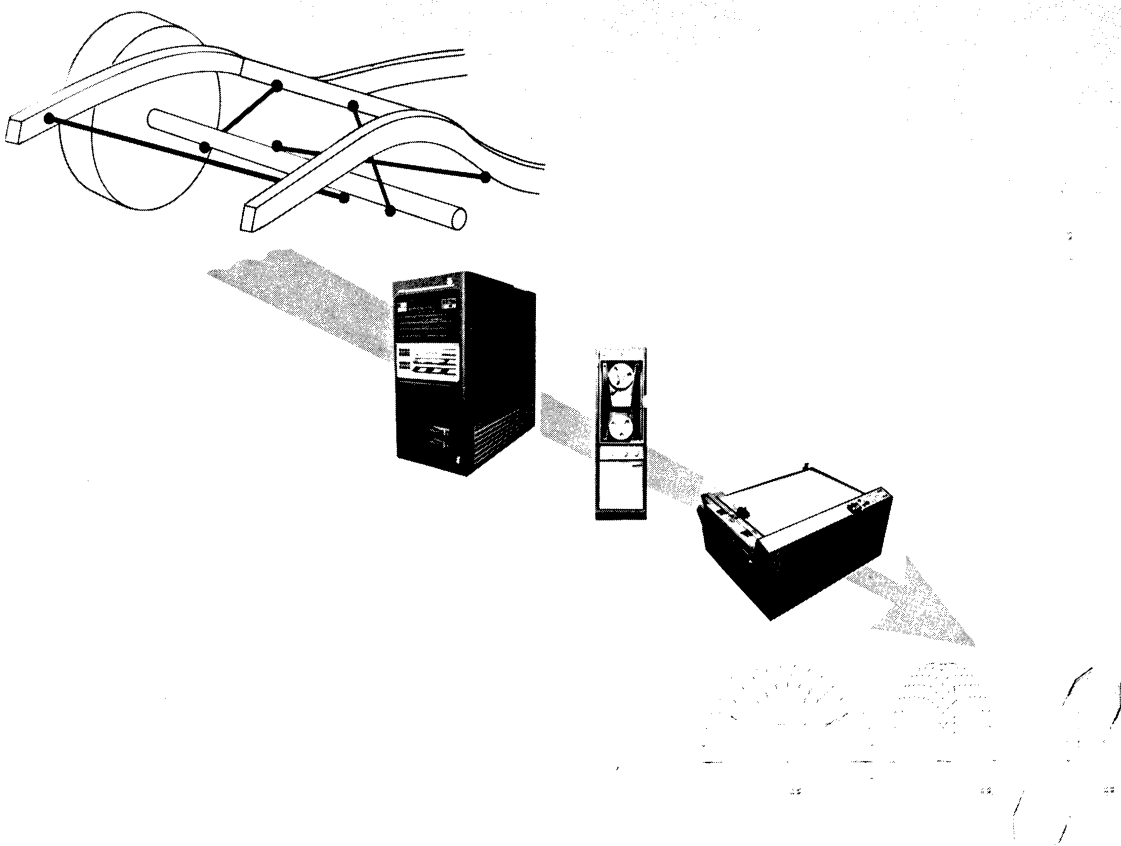


Figure 8

Figure 8 is the same thing again. The difference between this figure and the previous one is obvious. It simply shows the story from suspension designs through computer to tire plots and layouts.

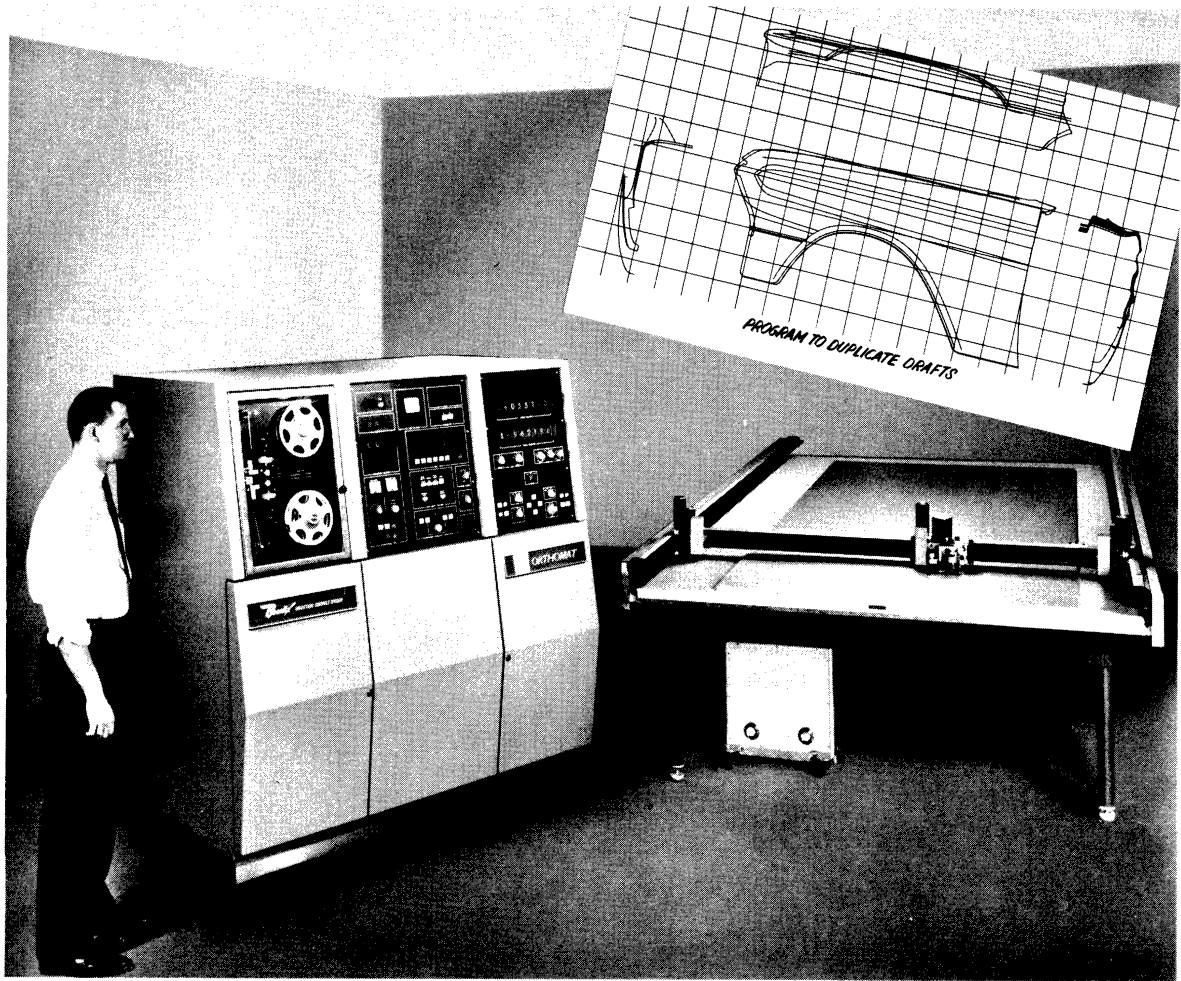


Figure 9

Figure 9 shows a typical large-scale drafting machine used in the aircraft and automotive industries for the output of graphical data from a computer in the form of accurate engineering drawings. I don't want to get off on the subject of accurate engineering drawings, but there are certain fetishes in the automotive industry, as elsewhere, about what is accuracy. Suffice it to say that these machines draw lines with an accuracy measured in the thousandths--much more accurate than most machine designers want, but certainly the kind of accuracy that people talk about.

**FENDER DRAWING
(COMPUTER AIDED)**

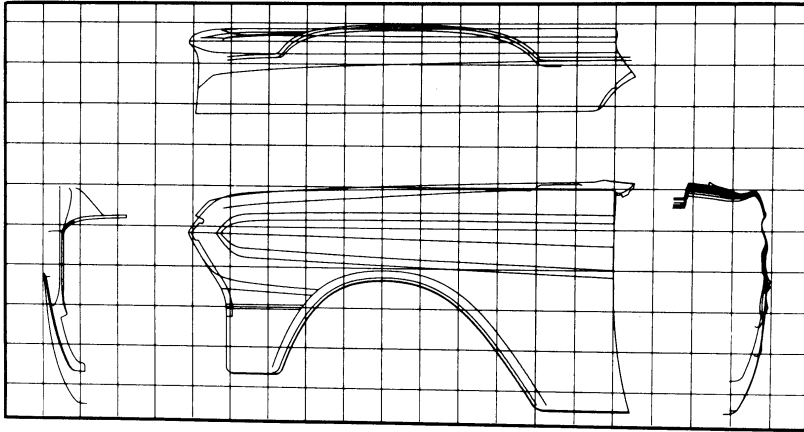


Figure 10

Figure 10 shows a large view of an automotive drawing, cleaned up for the purpose of presentation because otherwise it would be almost impossible to read. This kind of thing is produced on eight-foot by eighteen-foot sheets on automatic drafting machines by any number of companies.

Figure 11 illustrates another feature of using computers and graphics. For those of you who do design work and who are involved in things of this kind, you know what an arduous task it is to produce a lousy little perspective. It's so costly that you produce only one kind of perspective generally. This is probably one of the most trivial problems for a computer to do, and one that's cheap to produce. People are producing these things all over the place now in order to aid the designers. This kind of drawing can't compete with an artist's sketch, and it doesn't make very attractive copy for the car that you want to advertise but haven't built yet. (Artists, by the way, are reasonably easy to obtain compared to layout men to make these things.) Therefore, if you can design a car, or any other product, originally in the computer, or if you can put it into a computer, then producing perspectives such as I'll show you a little later on is a very trivial matter.

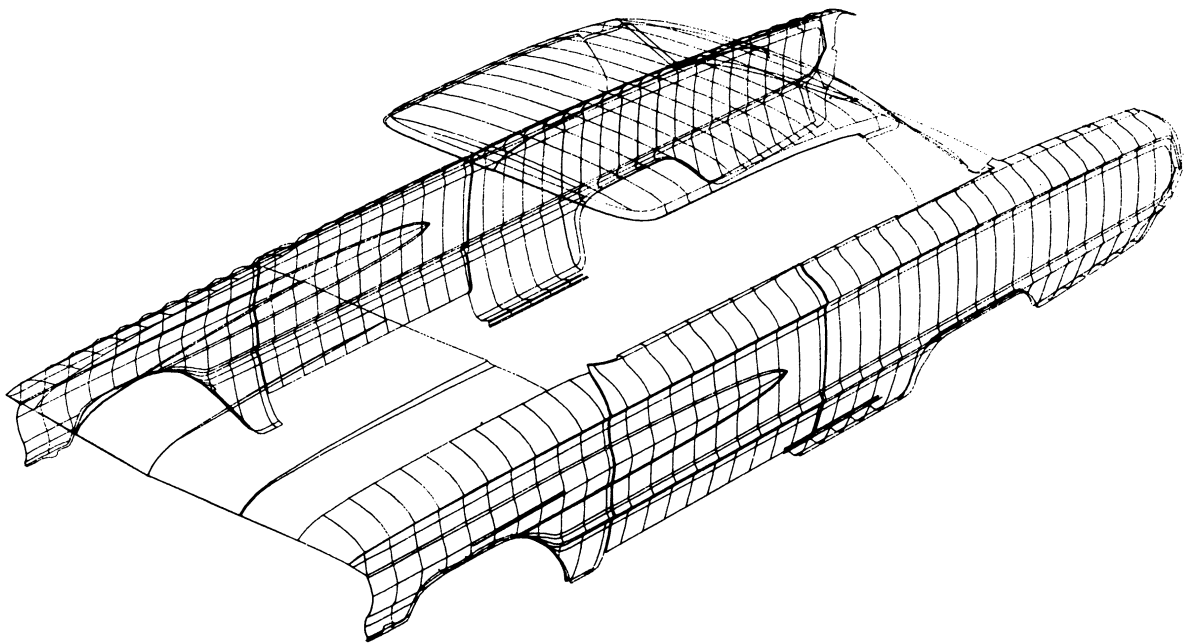
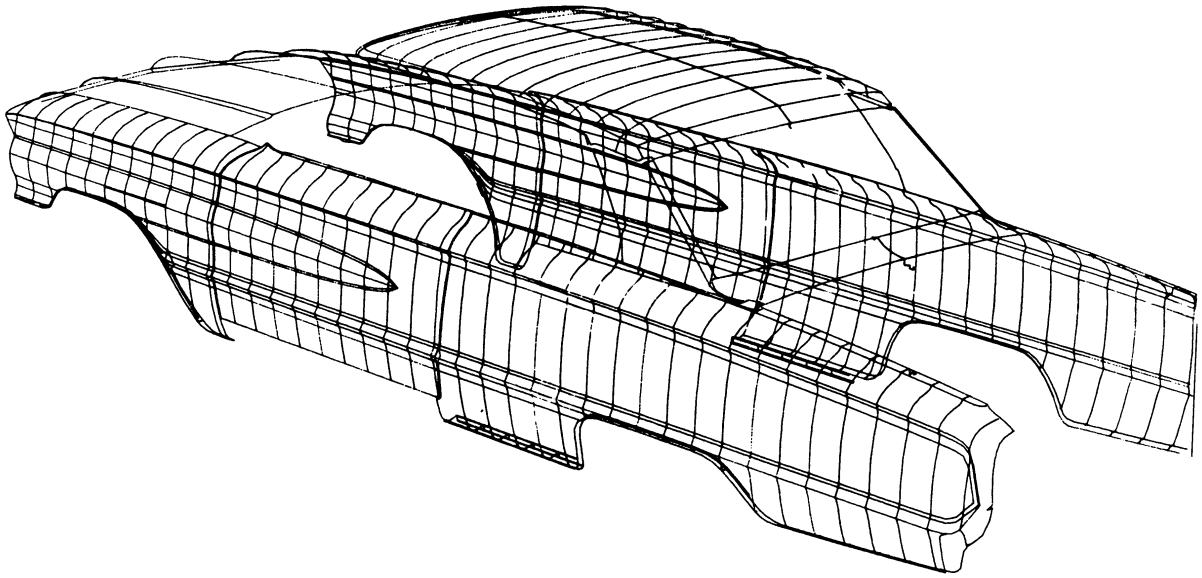


Figure 11

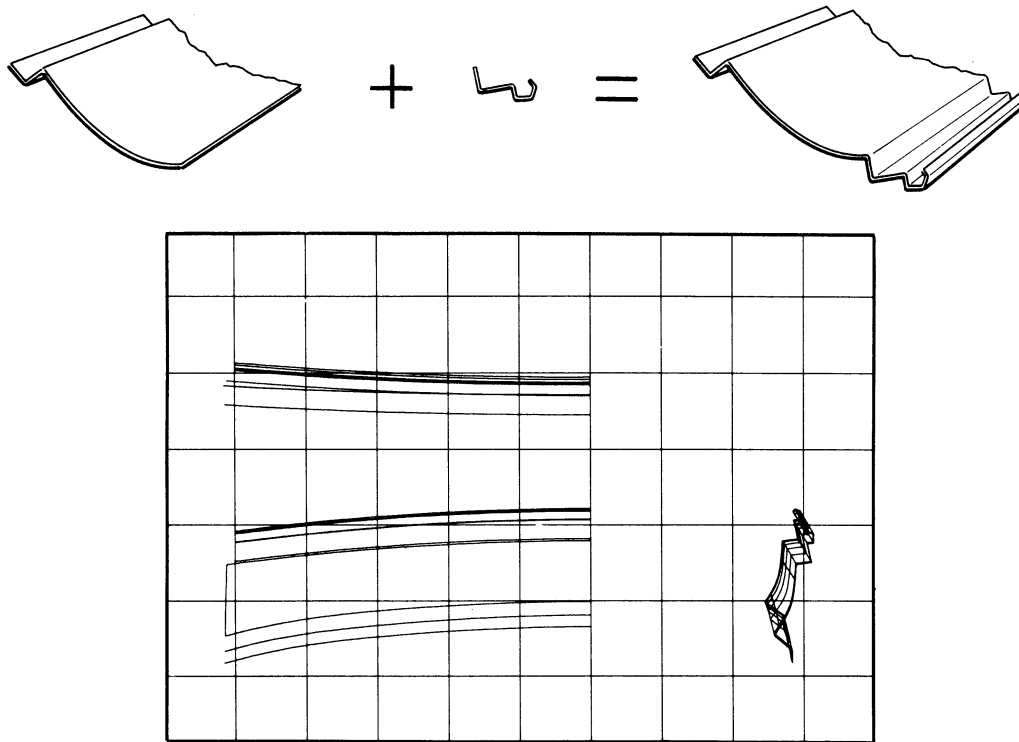


Figure 12

Figure 12 is typical of some of the things that we've talked about that we need, that the designers are doing, and for which we need, better facilities. This is the wedding cake problem. Last year's trunk leaked, and you want to put a new gutter around it with new rubber stripping, but you want to keep the rest of the sheet metal unchanged. I don't want to go into the economics of this, but it amounts to the following: You want a new gutter to run around the edge of that trunk in a certain precise way, having some relationship to sheet metal. Why is it the wedding cake problem? Because this is exactly what a baker does with a cake-decorating tube. He has the cake in front of him and a little pattern to follow. As he walks around, he squeezes the tube and lays the material where he wants it. The problem is as easy as that, except when you see it being solved in the drafting room. The man cuts sections, lays the thing over, cuts more sections, and throws it all back again. This is simply because he has two-dimensional data processing equipment for a three-dimensional problem. Intellectually this is not very acceptable, and mistakes are made all over the place. It is with this class of problems that one would like to see more man-machine interaction, but more of that in just a moment.

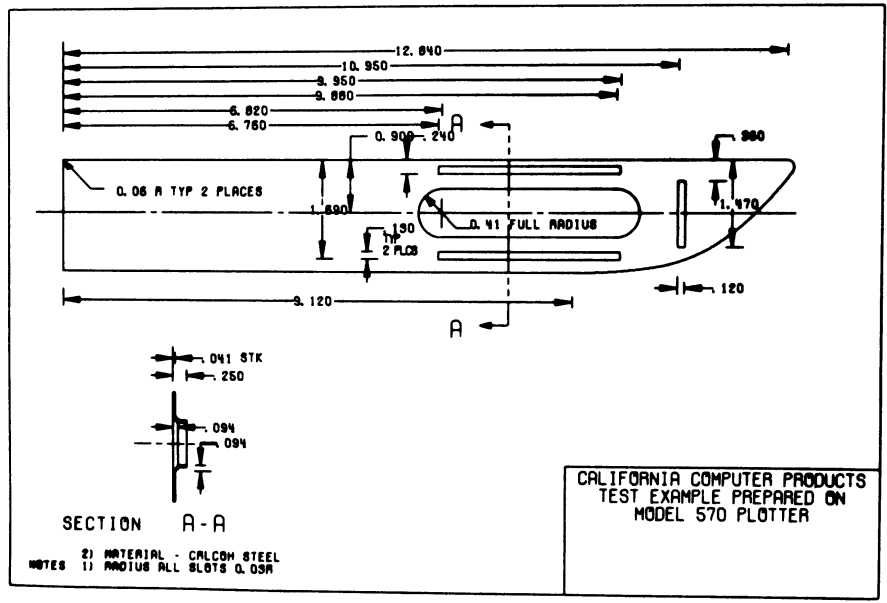


Figure 13

Figure 13 is a drawing made by a computer on a Calcomp plotter I bring it up here for the sake of argument. One of the problems facing engineering, vis à vis graphics and data processing, is that we cannot continue to simulate the way we do in business today and make total, effective use of the potential. But if your business is as production-oriented as the car business, for instance, you have certain trepidations about turning the whole applecart upside down overnight. It isn't a problem of being a day late with a building, or a month late with a building, it is the problem of getting to the market place on a given data. If you are not there you are wiped out.

If your car has been advertised as appearing on a certain date, and it happens to be a month late coming off the production line, you've had it. This creates a problem, typified by this drawing. Unfortunately, the photography of the slide is poor in that it has taken away all the bad features of the drawing. The plotter on which it was made can draw lines only up and down, left and right, and at 45-degree angles. And 45-degree lines do not make good circles. So you say, "I'll cut down on the increments." Even then it isn't satisfactory, unless the increments are somewhat less than the thickness of a ball-point pen, so you hide the imperfections in the smudgy line. Whenever this sort of picture has been shown to designers, it has brought up questions and problems about the relationship of designers to computers. I think there are serious problems and they need to be examined. It says on the drawing ".41 full radius," which is the machine designer's traditional way of specifying that thing that obviously looks like a circle. Arguments

always come out about that odd-looking circle, which is erratic because of the 45-degree lines and the up-and-down motions.

One question that must be carefully examined in introducing computer graphics and that does not come up in the kind of computing we are talking about is, "What is the function of the drawing?" One function of the drawing, to my mind, is to convey an idea or concept via a picture. What does the thing look like? If you are the production manager of a remote plant, you do not care about how many thousandths offset there are; you want to recognize an object and talk about it. But the man that produces the part had better know precisely what he wants. With a compass, it is easier to draw a good-looking circle than a poor-looking circle. So we always draw good-looking circles, and then we redundantly say, ".41 full radius." That is the full impact of that message. Therefore, if you ask of computer-driven displays the same kind of prettiness of drawing, you are asking for a system that is difficult to justify economically. On the other hand, if you recognize that this picture, however crude, also contains precise information, then the system needs no justification. Any of the common computers have a lot more accuracy than you could possibly need to describe any reasonable part. So this drawing, to my mind, should deserve some attention from engineers as they look toward the future.

Figure 14 is from General Motors and shows the DAC system room. On the right-hand side, you can see a cathode-ray-tube device at which a man can sit and can do keyboard manipulations. He can point with a special kind of pointer that the computer understands, and he can do other things which you cannot see because the buttons are hidden. Toward the middle is a machine with which I, personally, have a philosophical quarrel. It is a rather important device in the sense that the technology is somewhat better. It photographs a drawing on 35-mm film which can then be scanned with cathode-ray-tube devices, photo diodes, etc., and the resulting information can then be stored in a computer. Philosophically, it is essential that in on-going processes one have facilities for putting information into a computer from the existing drawing. Whether this particular way of doing so is a good one or not is somewhat immaterial. This one, because of its small size, has limited use in a world where the drawings are much larger. As an engineer, I would attack some other issues as well, but as a reporter of computing devices I'll refrain from doing so today.

Figure 15 shows a Chrysler employee at the General Motors DAC console; it gives a more detailed view of the kind of things one can do. The pencil, or pen, that he points with is one way of communicating with the computer. The remarkable thing that I noticed, in a

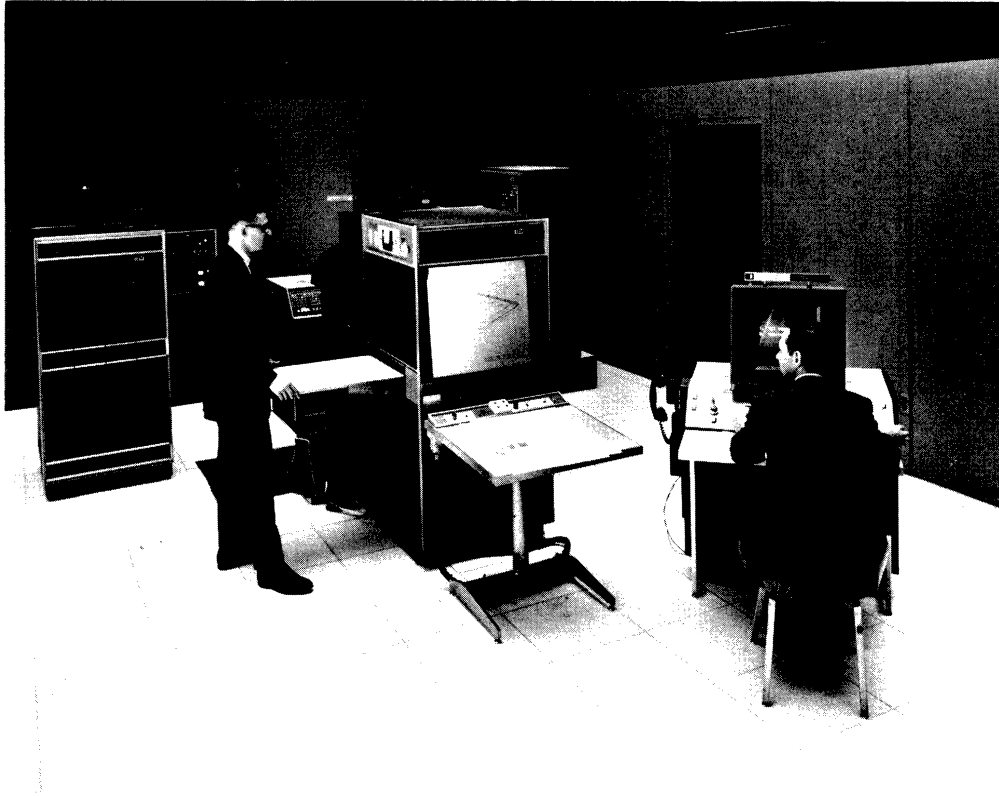
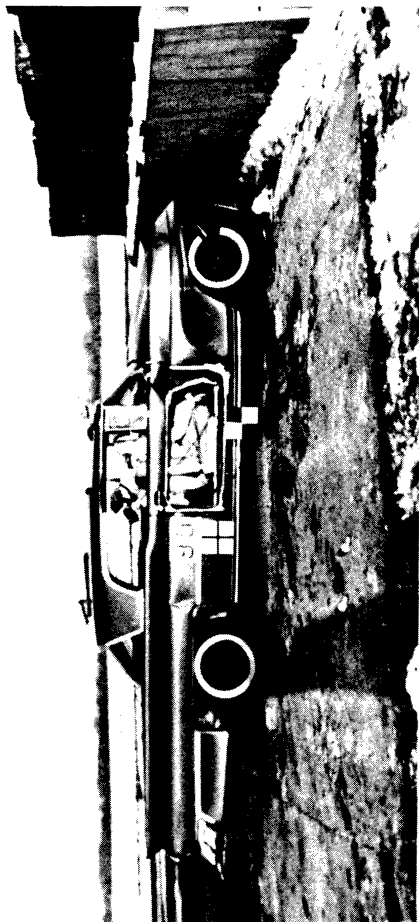
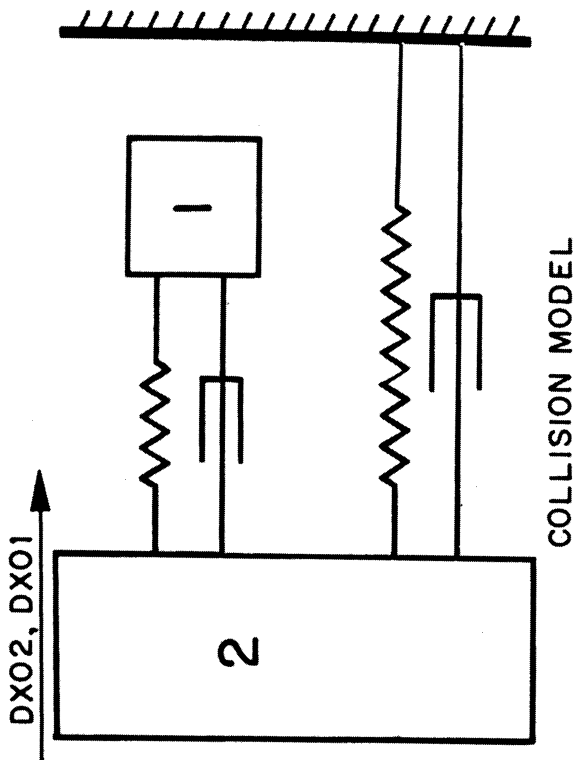


Figure 15



```

C DYANA I-TRANSIENT MODEL FOR VEHICLE COLLISION
X SYSTEM DESCRIPTION
X EOOM01, EOOM02, EOOK02, EEOC02, E01K02, E01C02
X SPRING RATE, EOOK02
X EOOK02 = A + B*X02
X DAMPING RATE, EEOC02
X EEOC02 = C + D*DX02
X READ INPUT VARIABLES
  A, B, C, D
X TIME DEPENDENT COMPUTATIONS
X FORCE = E01K02* ( X01 - X02 ) + E01C02* ( DX01 - DX02 )
X PRINT-PLOT ANSWERS
  X01, X02 VS. TIME
  DX01, DX02 VS. TIME
  DDX01, DDX02 VS. TIME
  FORCE VS. TIME
  X01, VS. DX01
X END
    
```

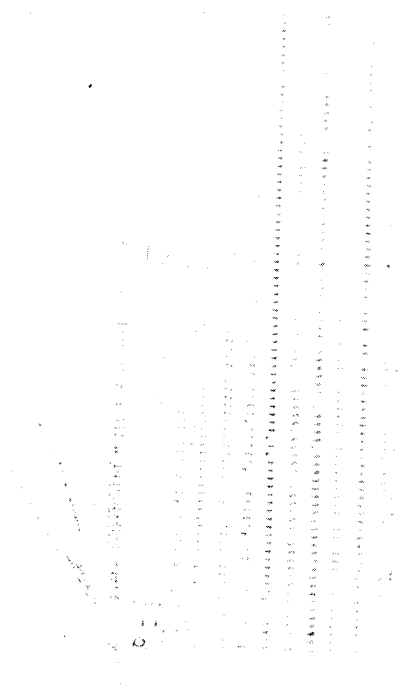


Figure 16

rather casual observation of this system, is the ease with which traditional designers use it. I've seen people use it without a plastic overlay for the keys. They knew what those buttons meant in the process of solving the problem. On occasion I have seen people point at the screen giving the answer before the question was asked. This shows you something about the adaptability of people to a system. It may also say something about the slowness of certain parts of it, but I don't think that's very significant here. The important thing is that people are adaptable to the system.

Figure 16 was made before the recent safety trend; it shows several things that should intrigue a designer. I was trying to demonstrate how the safety engineers were using Company A's computing system to solve Company B's problems. On the left is a picture of one of the many periodic tests of running a car into a wall to see what happens to the dummy inside. The mechanical engineer's version of that picture is in the upper right. It consists of some blocks, springs, and dampers. This was the simplest slide I could put together showing the features of the problem. In other words, 1 is the driver, 2 is the car. The squiggly things between 1 and 2 represent the seat belt and the junk at the front between the wall and the car, 2, represents what happens to the front bumper on impact. The system, call DYANA, allowed a rather simple format of language to describe this problem. It talks about E00, M01 (M being mass), etc., which was useful in describing a problem of this kind, without being a programmer, to a computer and using it. Quite obviously I could sit down at the teletype and introduce such language to the computer and ask for results. In this case the picture shows cards, but I would like to go one stage better. I would like to stop at the upper right-hand picture and say, "That's what I want to talk about," and have the computer understand these pictures, which are graphical in nature now, schematic, if you will, and then start asking me questions. It might say, "This looks like an interesting system. What is the value of mass 1? What is the value of the spring between 1 and 2?" I could then sit down at the teletypewriter, or some other device, and introduce the appropriate numbers.

Figure 17 shows one of the potentials that was obviously realizable years ago to make such a computer program--a more interesting model of the collision--and have it translated into a picture sequence that could be used in an animated motion picture. I will show you a strip of such a movie in a few moments.

A system that we cannot pass by is the Sketchpad system at Lincoln Laboratory at MIT. Figure 18 shows a demonstration of it. This system is contemporary with the General Motors DAC-I, but because it was not specifically oriented toward an engineering product, the comparison will point up some interesting, distinctive features. Again, there

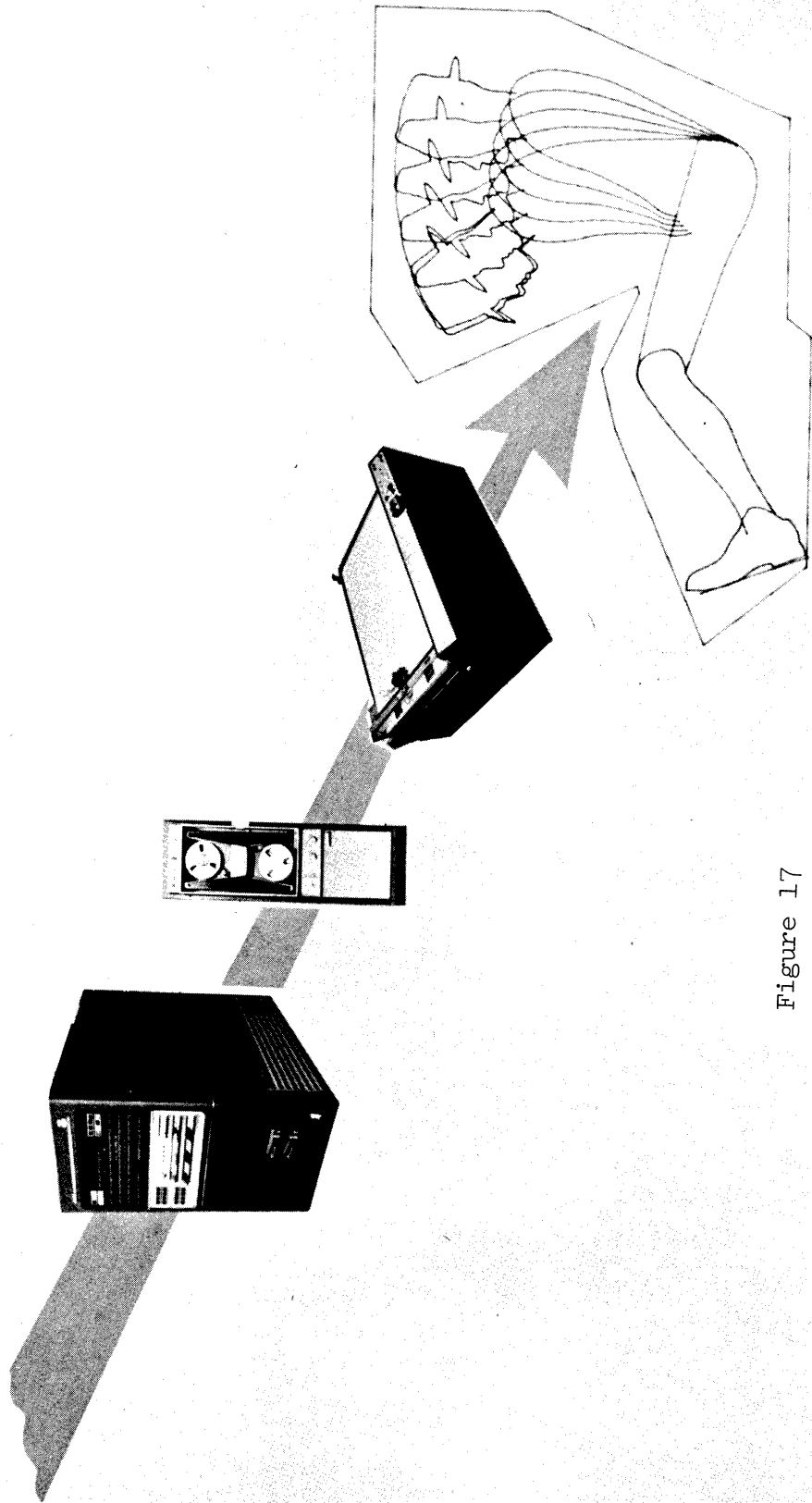
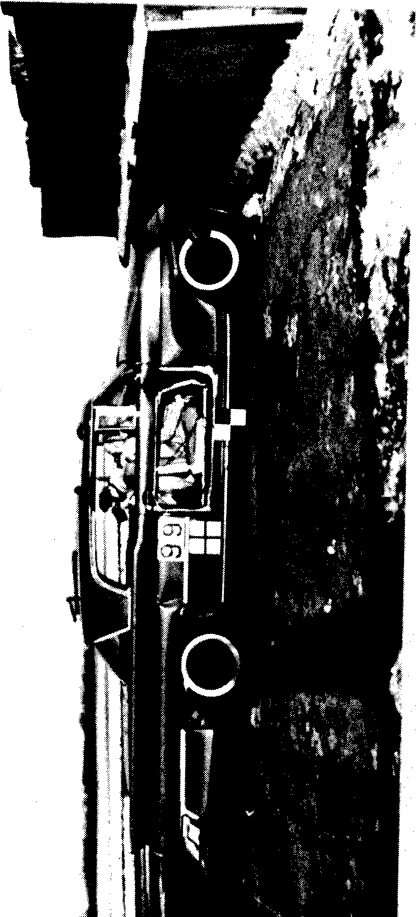


Figure 17

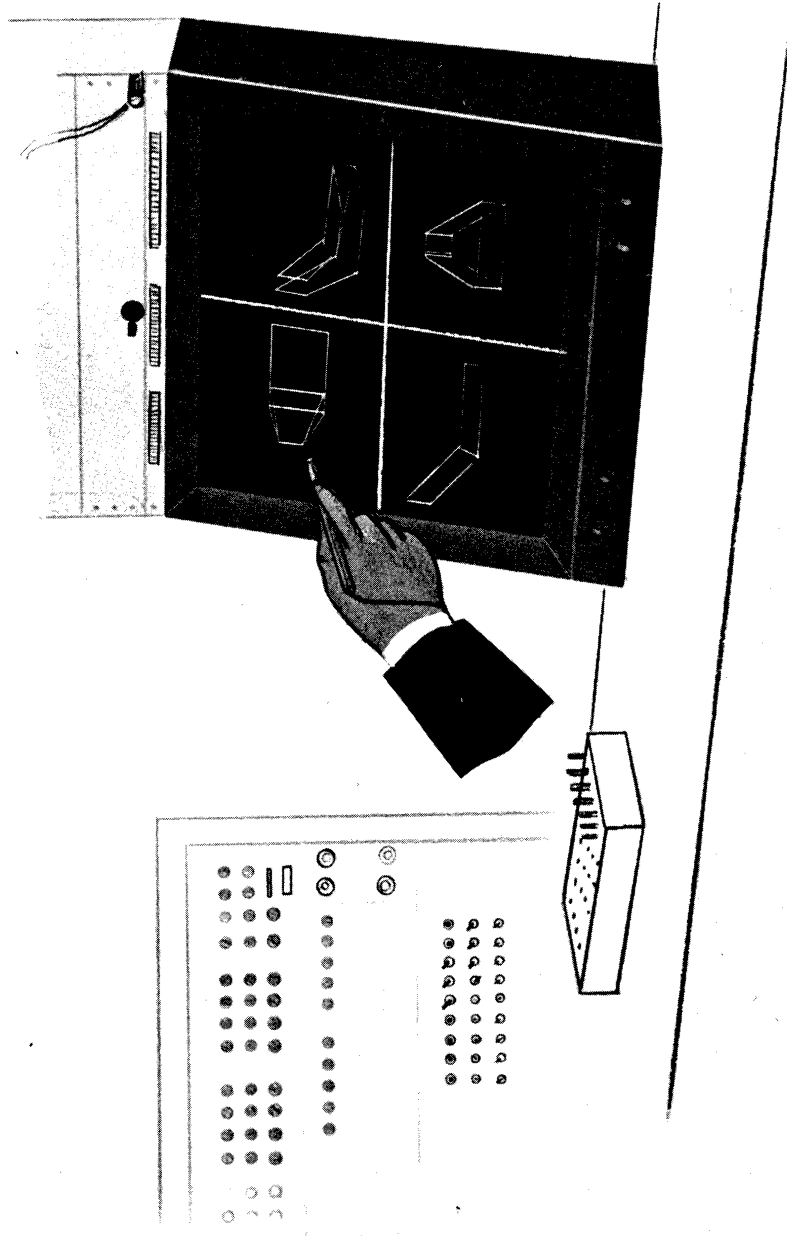
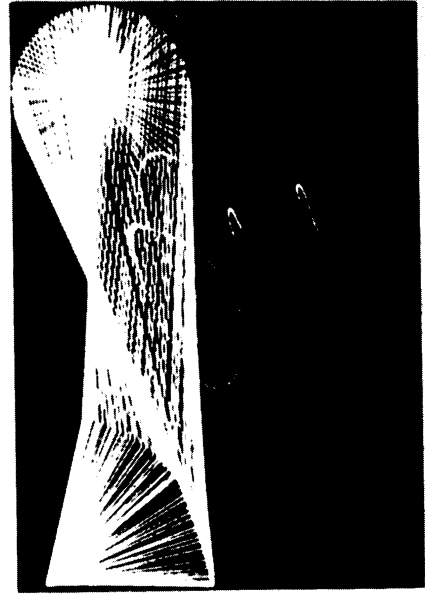
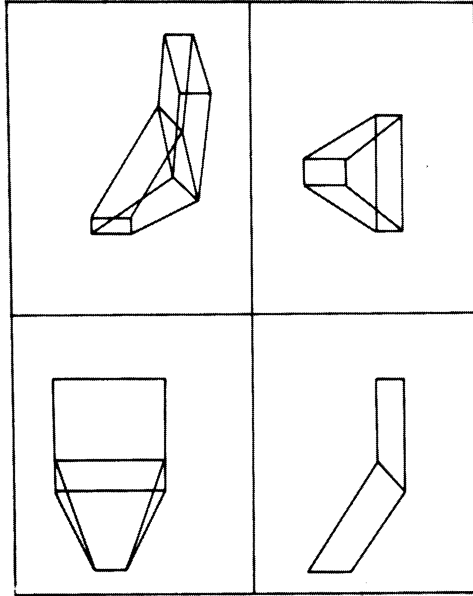


Figure 18

is some kind of cathode-ray tube and some kind of a pencil, called a light pencil in this case, with which you can interact with the computer. In fact, you see that the picture displayed on the CRT screen (and reproduced for clarity in the upper right-hand corner) is the traditional orthographic view of an object accompanied by a perspective--there is an important difference. Were I able to show you films of this, you would see that when one is drawing in one view, all the corresponding data belonging to the associated views are immediately introduced. This is precisely what you like to do when you make an orthographic drawing except that you are the computer and you do the correlation. In this sense we have taken out the computing part, the busy work, the bookkeeping part, that falls to the lot of the designer and is in no sense creative; in fact, it can be very destructive if he does it badly.

The lower picture shows a mechanism problem--a very interesting one in light of the window-mechanism problem we talked about earlier. On this computer it's entirely possible to draw lines, instruct by pointing, judicious pushing of the buttons or various other techniques of talking to a computer, and say that this point will be fixed, this line will not change length, etc. What I've done in this geometric interplay is to describe what engineers call a four-bar mechanism. The computer comprehends these commands. When I move one part of the mechanism you can see a complete outline of all possible positions of all parts of the mechanism. These are, therefore, valid engineering points, and you're tracing out here the path of a particular point. For this type of problem I don't have to tell a computer all about kinematics; they are described inherently in the system. This again leaves a large degree of flexibility in the interaction between the computer-user and the computer. It requires considerable computing power and it requires considerable sensible intelligence. Practical mechanisms tend to be much more complicated than this, and you frequently make untrue statements about them. When you try to move the thing it won't move; or conversely, when you move parts of it others don't move, which is a question of whether you over-constrain the problem or under-constrain it. These things would show up rather readily in this case.

Figure 19 shows some doodles I made on that computer. The occasion that I'm simulating is you attending this talk. You've found a paper clip in your pocket, and because you're getting restless you start bending it. "Wouldn't it be nice," you say, "if I had these three orthographic views and I could pass a smooth curve through the corner points of it?" You sit and dream about this (people actually do such things!) and the curve in Figure 20 is the computer simulation of your daydreams. (I'm trying to nibble a little bit at the esthetic problem without hitting it head on.)

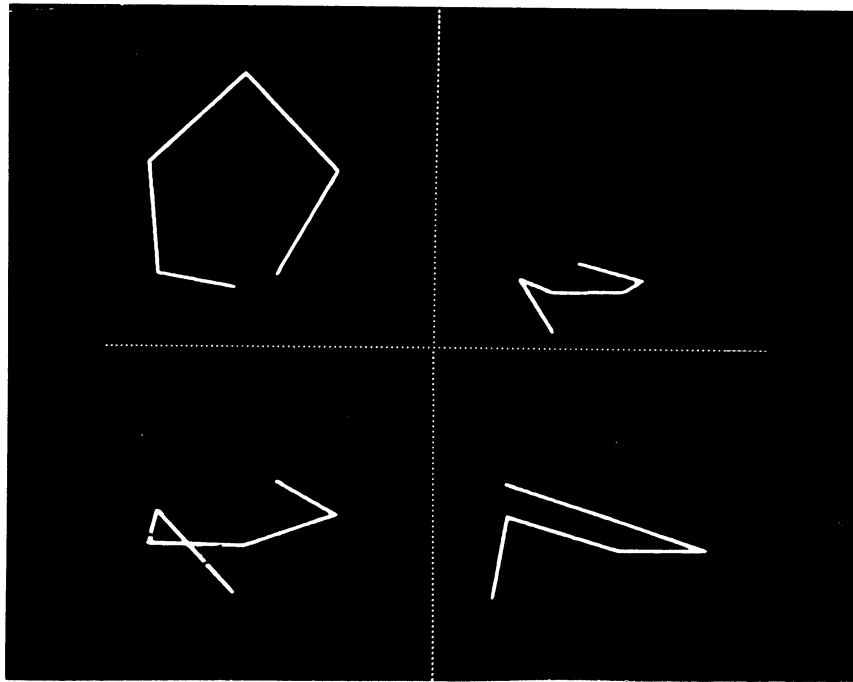


Figure 19

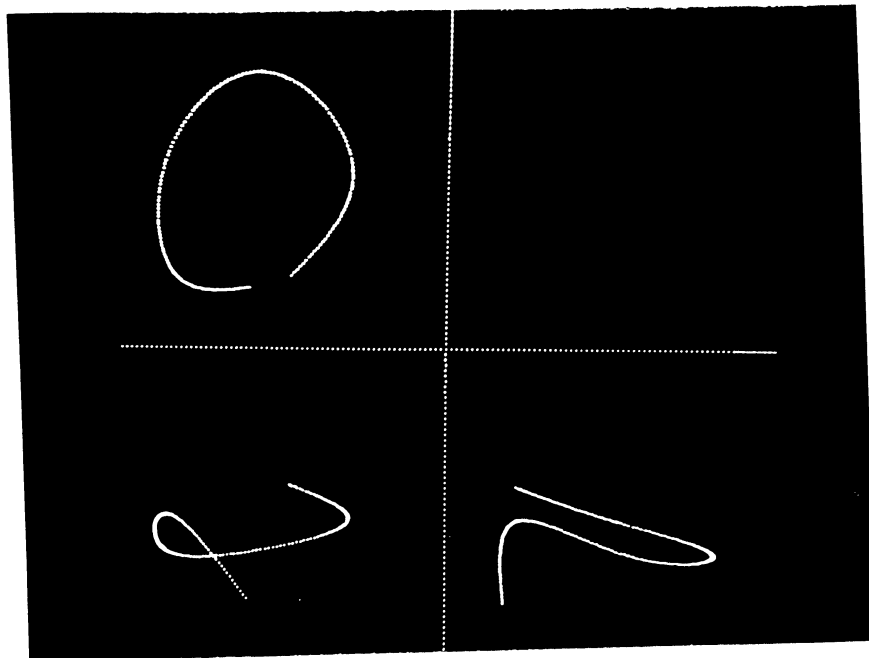
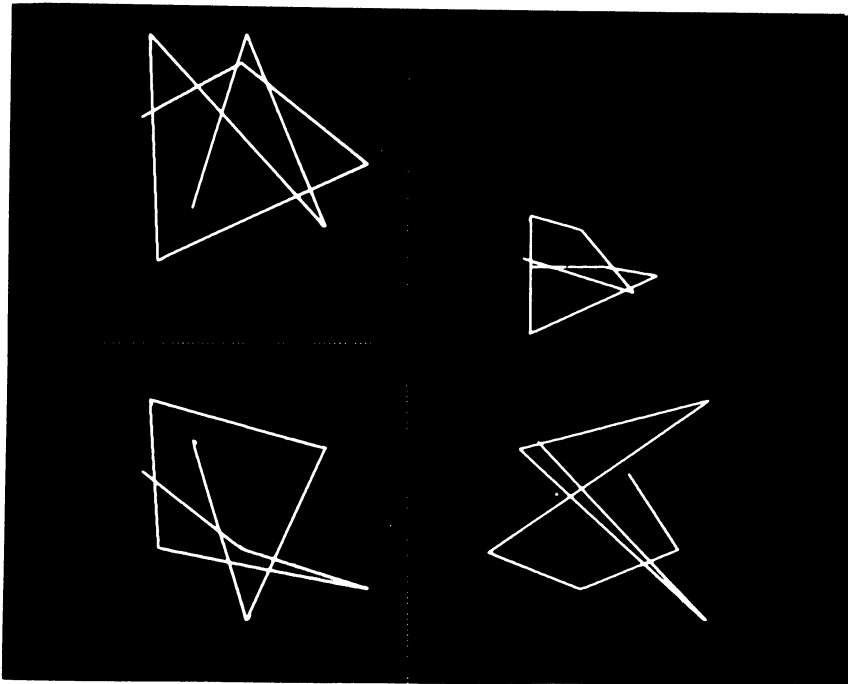
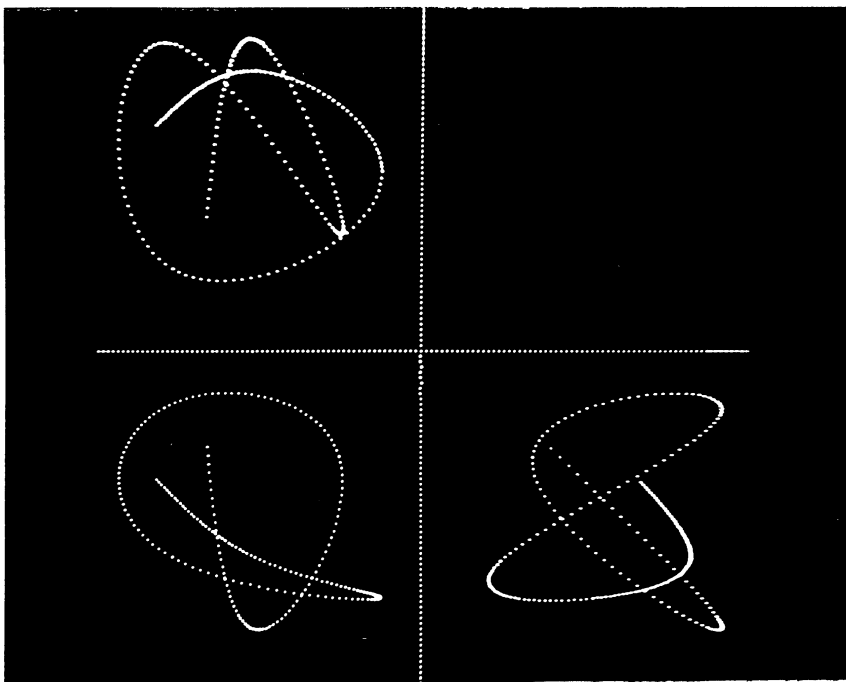


Figure 20



(a)



(b)

Figure 21

There are great collision courses between styling and engineering, and between architects and engineers. I'm happy to say that about 20 architects are now taking a course in computer graphics and that peaceful coexistence is quite possible. The problem facing many people who deal with product design and many engineers who have to come up with an esthetic shape--this is keenly obvious in automotive design--is that certain essential engineering features must be carried through--wheel base, wheel height, engine, and so on. The automotive designer wants to enclose his product in a skin that won't reflect light in a bad way in the showroom. He could care less what happens when you drive it home and your wife creases a fender against the garage; you've been sold already and you can go from there. But to sell you that car, he wants clean lines. With airplanes this is not the case. I saw some Ford people, body-engineering types, walk past the Boeing 707's one day in the plant. They looked up at these monsters and said, "We could not sell one of those, look at that skin--how wrinkled it is!" For aerodynamic purposes that wrinkled skin, due to riveting, is of no consequence. The essence of the problem, and of the fight that I see between the stylists and the engineers, is that, in fact, engineers with computers will win this game. You can instruct a computer as to the nature of the microscopic features of that surface, which you cannot do by hand. I can give the essential shapes that I want (last picture I showed), hit a button, and get output like this. Do you like it? That's fine. If you don't like it, let's change it; but one thing I assure you, the surface or the curves so generated have the characteristics that you want. This is something that I cannot expect you to do in a sketch or a drawing without great labor. Frequently these are very simple characteristics, although I suggest that there are yet some areas to be disputed here.

Figure 21 shows a more complicated version of this. It would be nice to be able to turn that picture around into various views and to see what happens.

I'd like to show you a short strip of film and then close this talk. In this film you will see two sequences: one is an example of the safety issue that we have just discussed, and the other shows something about manipulating surfaces with a computer (Figures 22 and 23).

I've shown you these various things to give you some idea as to the kinds of things I think a designer should expect to do.

Asking again, "What are the consoles of the future?" I started out by telling you that I don't have the credentials to tell you about each one, because every time I go to a convention I hear of another Rube Goldberg idea that has great appeal. But let me tell you about some of the types of things that are available. The cathode-ray-tube devices

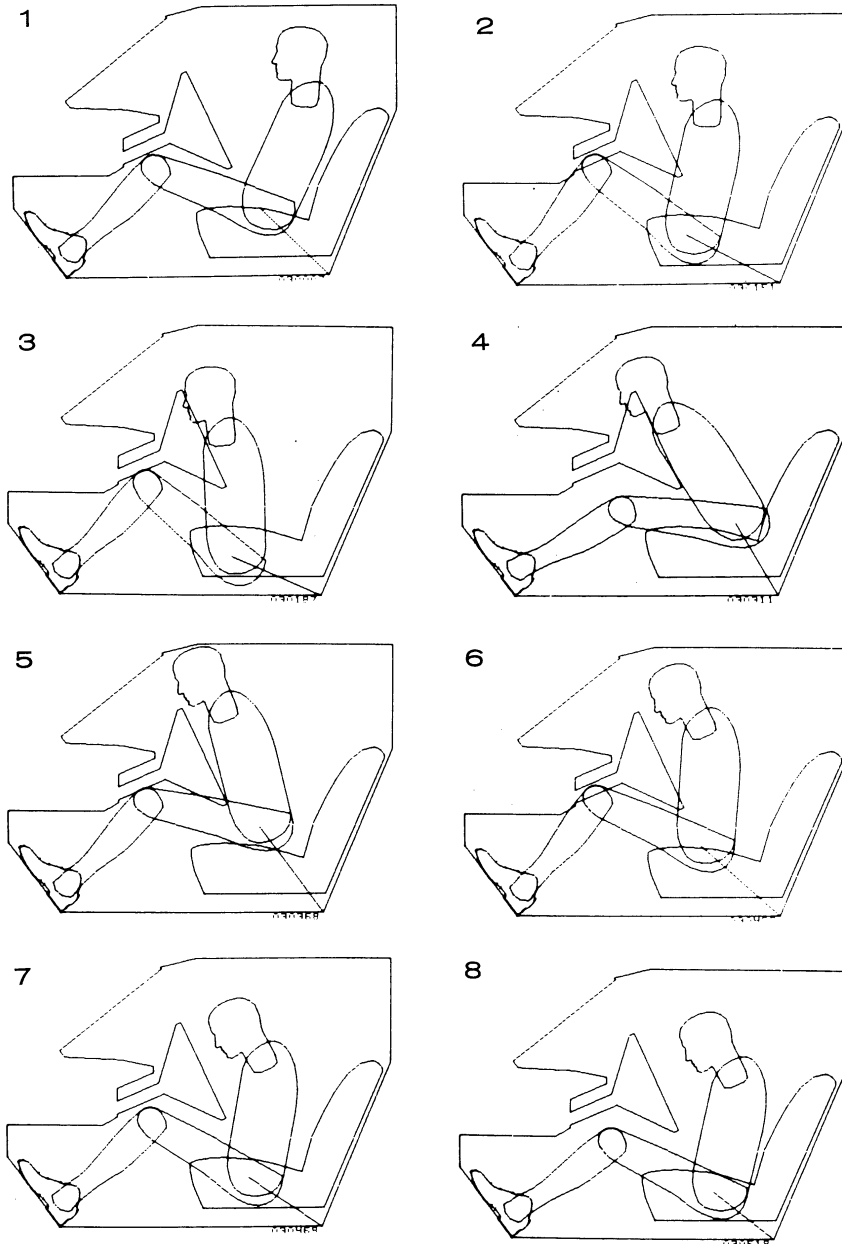


Figure 22

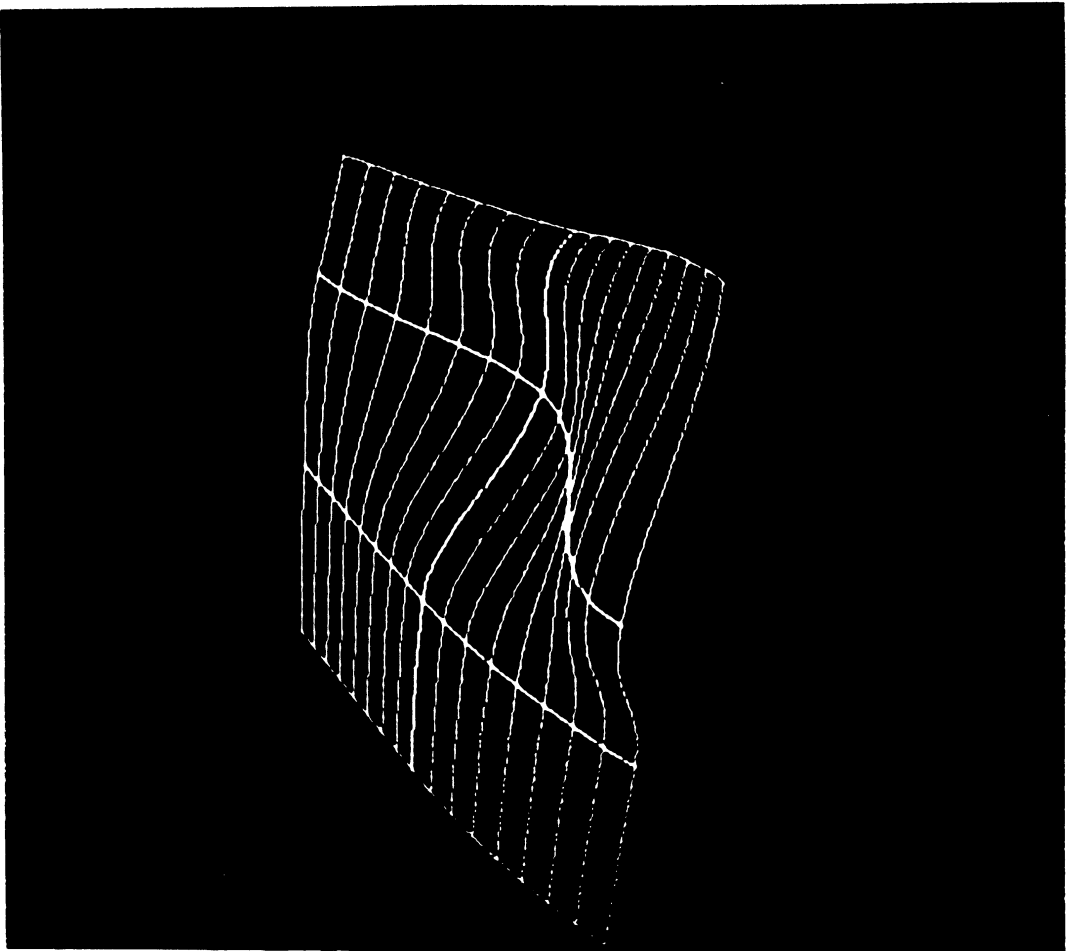
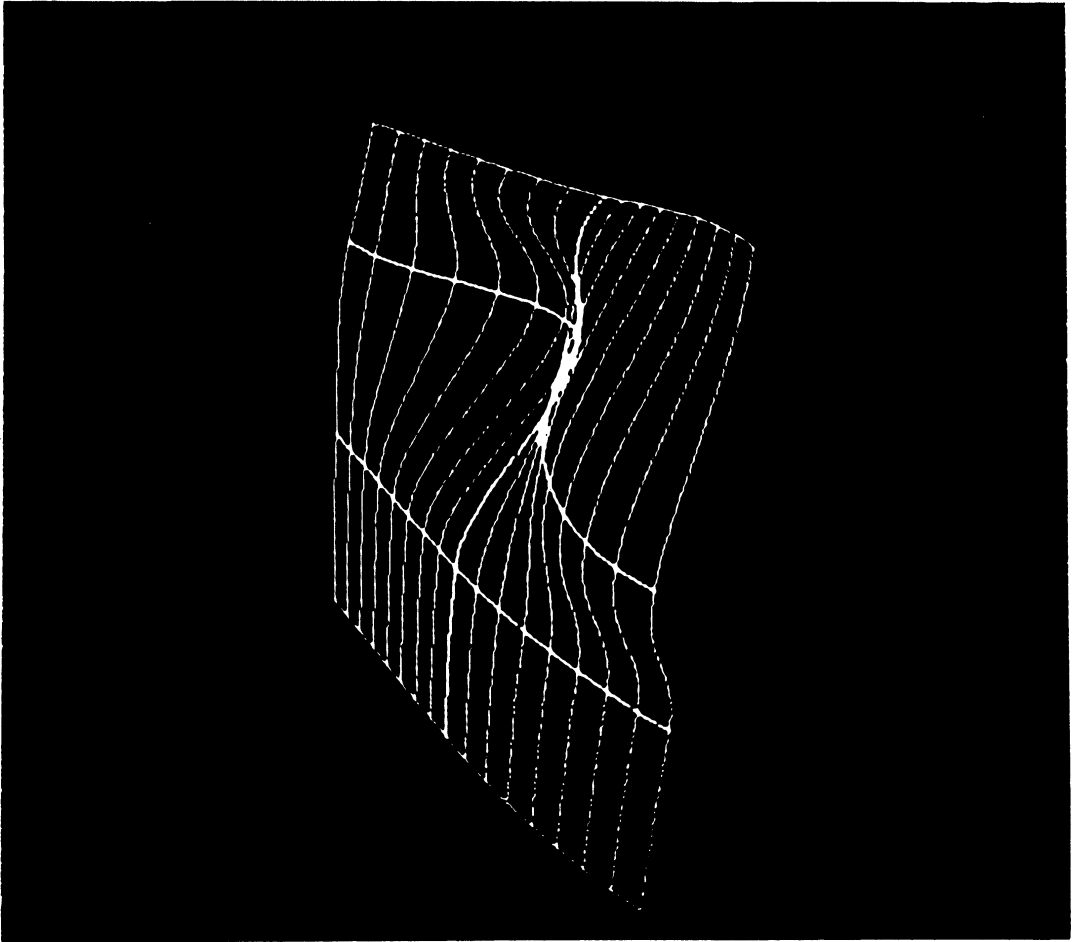


Figure 23

that you saw in some of these examples are very expensive today. They involve a costly computer investment. The present generation of computers is presenting some interesting challenges by way of reducing this investment. In graphics, however, we have not reached the stage comparable to that demonstrated today by the ease with which I picked up a Dataport, trundled off to the nearest telephone, called up a computer, and talked to it. By the same token, however, we are not as ready to use such devices as we might be. This is a challenge for engineers, not for the computer people. Basically the hardware is there as someone said earlier today. The demonstration, the use and the implementation of these things still require a lot of work. They require a lot of enthusiastic work by engineers.

We're becoming a timesharing world and all these devices will ease this problem, but it is still a very costly problem. However, at Santa Barbara, a man named Culler, with his colleague Fried (Culler used to be a Michigan man), developed a very cheap cathode-ray-tube device with which he can do his favorite problems, which involve the teaching of mathematics. He can describe functions and see what they look like, plotted immediately, and where the interaction goes on in this sense. Devices are becoming available, essentially out of MIT and a few other places where there is some development. We are looking for \$5,000 to get a cathode-ray-tube terminal that you can hang onto a telephone line and on which you can display some pictures, however crude. These things are stored in a delay line, for instance, and you can see things and get visual information. I think this is significant, particularly in view of the comments I made about the information content of a drawing as an idea transmitter in contrast to a dimension transmitter. To get dimensional data you have only to ask the distance from one point to another and identify the points clearly on the cruddy picture. The computer knows, symbolically, which points those are and can find the dimension precisely. The application is obvious. When an engineer wants to put on a flow diagram or when he'd like to put on a mastering damper, accuracy is not at all the problem but broad picture capabilities are important. We're looking to these, but you'd still like them to come down to \$1,000, and you'd like them to have better capabilities.

There are things like the RAND Tablet, which is essentially a sheet of plastic with all kinds of wires beneath it on which you can write. People are able to recognize writing here, in a limited sense, and on a cathode-ray tube you get a cleaned-up version of your writing. I've used the device, it's not infallible, but essentially the people were using it in programming. What I'd like to do is draw flow charts; never mind all this programming. These devices will make some of the communication easier.

Another man (Doug Englebart of Stanford Research Institute, who's a particularly clever man) has a bug, as he calls it, which he holds in his hand and moves over the table to indicate a motion in one direction or another. But when does it stop? Well, the picture stops where he wants it to be. And he's a particularly clever man, because I'll be darned if I'll ever be able to do it, but with five fingers he can manipulate an alphabetic code. Obviously, with ten fingers and three positions on a typewriter keyboard, 26 can be coded into some configuration, but he's clever enough with his five fingers moving back and forth. I'm not sure how many keypunch girls you can convert, because its obviously a rather personal type of device. But if you look at what some of the learning people have done to master some of these devices, maybe there is a happy ground in between.

In contrast to the light pen, which positions you in the plane, Lincoln Laboratories has an interesting device--something they call the "wand," which is a three-dimensional device. It has microphones and transmitters, and essentially is a little radar-type device. It sends out signals to determine its location. You can move it around any place within a four-foot cube, and that position is relayed to the computer. One of the interesting ways of using it is to use the position away or toward the console as the information for enlarging or contracting the picture.

A demonstration the other day showed something about making things economical. Pushing information out of the end of a cathode-ray-tube is very costly, given the speeds of cathode-ray tubes, the speeds of computers, and the speeds of communication lines; so there's a real problem in transmission. Next to the computer was a printed page with pictures defined on it, and the position in space of each picture was given as a coordinate. This coordinate was understood by the computer and referred to something stored in its memory, so that you now could refer to a much larger repertoire of symbols. In fact you might make a house from a block, a roof, and something else and call it a new symbol. You don't have to crank it out of the end of the tube and look at it to make that decision. If you're in a parts assembly business where you have a standard catalog of parts, you put these parts on a picture, mount them on a specific board which has a coder at the bottom and, as you buy these parts, it turns another code. The computer knows by that turn and by your position on the page what you are calling up. Now this is obviously bringing us back down to dirt-cheap media to get this information.

A year ago at the Fall Joint Computer Conference an interesting discussion started which I think is significant in trying to put some dollar values to the future. Some people started out with the new micro-electronics, and said that you can get so many logical elements on so

many thousandths of an inch, and so many thousandths of an inch cost you so many pennies to produce. You can't get one, you have to get a great number of them because that's the way they're produced. The suggestion was that I could have a desk in my office with a cathode-ray-tube terminal which would have the computing power of a 7094, and the memory and everything else with it right there in the desk. The biggest complaint was that the technology of memory was lagging and that it cost a lot of money. This you should be able to have. They were talking about the future, quoting 1970 as I recall. You have to bring it down to earth a wee bit. In my mind there are certain hesitations about accepting a \$30,000 investment as conservative, because the computer business is having some troubles too, but on the other hand this is possible. But think of what potential computing power you might have. It will solve some of the communication problems.

A man named Hopplebaum, the year before, sat at a conference and said, "One of these days it'll be possible for me to sit at a console wearing a helmet like a miner's cap with a light beam of some kind and start thinking about an idea such as a butterfly. I would sit there with my hands in my pockets and by moving my head I would eventually generate a picture of a butterfly on the face of the tube." It's interesting to note that Ivan Sutherland, who has a leg up on this business, is now at Harvard University working on something very similar to this in the form of a helmet as a transducer to see what's going on, and observing in some sense some of the eye movements that you carry out to see what this relationship is. These things we hear about do materialize.

Designer is a very broad word and I use it very widely. Figure 24 is a picture of the United States as a population distribution. You'll recognize immediately the location of New York City, Los Angeles, and San Francisco. I think it's a rather interesting picture, done by a geographer with the computer and plotting devices, and far superior to a census book in conveying the idea of population distribution in the United States.

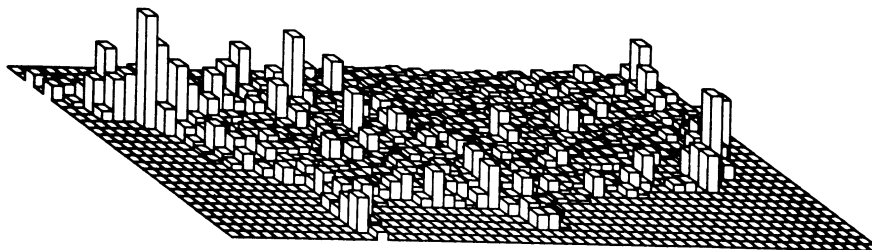


Figure 24

COMMUNICATING WITH COMPUTERS FROM REMOTE POINTS

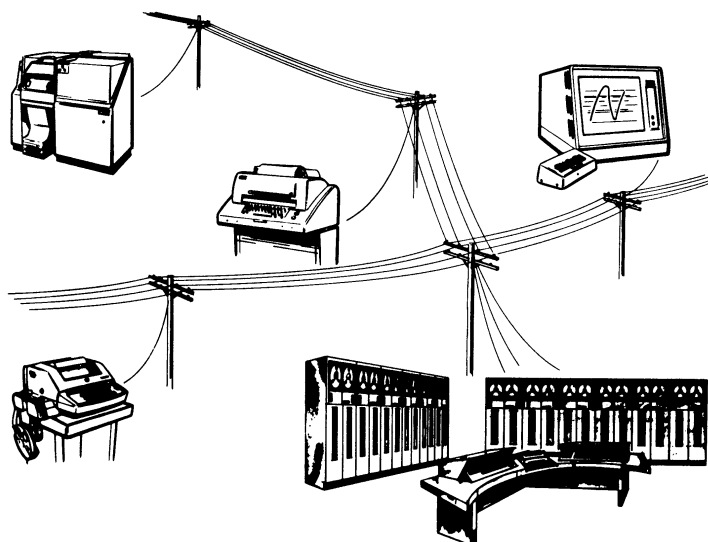


Figure 25

Telecommunication is the problem in Figure 25. If you can do graphics on a computer, the engineer will want to use the terminal where he works and not have to go over to a computing center, as he's forced to do today. Then the question is: What does a terminal look like? Yes, it has buttons on it, it has graphics on it, it needs some memory to refresh the picture, it needs the sending of information back and forth. Information transmission is difficult because nobody anticipated the growth of high-data-rate transmission. It is expensive because of archaic structures of equipment, and policies, and the telephone company; many of you already face this problem. The question, therefore, is: How do I divide my work between a large computer, which I need occasionally, and a remote terminal that is comfortable and graphic?

Figure 26 shows some equipment we obtained recently, at a cost of roughly \$100,000. The project that we're working on, sponsored by the Advanced Research Projects Agency of the United States Defense Department, is investigating the problem of the division of labor between the remote terminal and a rather expensive terminal. This is, after all, an appropriate research environment and if we wait until the terminal is down to \$5,000 then another whole era of research and engineering has gone by and not research and computing. How small should this remote terminal be? And, how much should it cost for me to have a small, remote terminal, with access to a large, central computer when necessary?

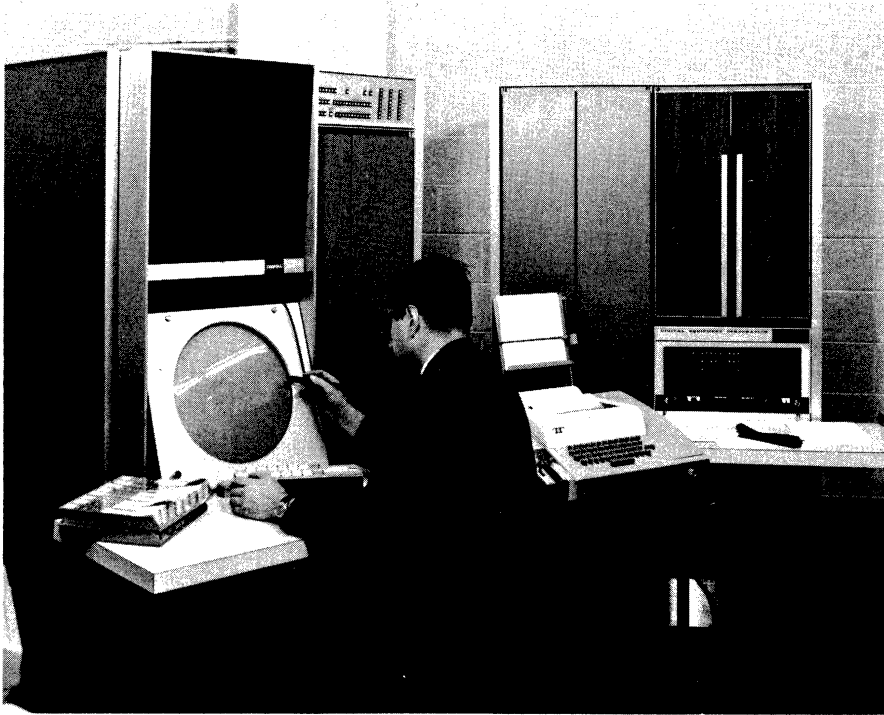


Figure 26

In the environment of remote teletypes and graphic consoles, people today wonder how they can manipulate certain things on a computer without having to reprogram everything. There are many interesting experiments going on across the country in an effort to hook up to a computer on the West Coast, say, only when the programs on the West Coast are needed, rather than to call up the West Coast and have the program sent over, only to find that they have a different configuration or even a different make of machine. We suggest that you utilize the computing power and communicate with it. I think this is an important piece of work.

There are three things that make the future brighter: get these terminals defined; get the division of labor defined; and get some cooperation between various facilities. I recognize, particularly in this audience, that there are certain proprietary competitions going on in this area in industries like the auto industry. I think, however, there has to be a clearer decision about what is proprietary in terms of a product or marketing posture in comparison to something that's proprietary in engineering. (I don't think we generally patent a course in engineering mechanics, for example, because we've got a leg up on it.) This is something I hope the future will bring with all kinds of terminals. I failed really in a prognosis of the future, but I hope I've painted a picture of some of the things that are here and that I think are coming. Thank you.

DISCUSSION

Question: What does the Dataport cost?

Answer: This thing is around \$2,800 right now. It's still over priced and it's still a little bit heavy.

We have a real problem with the telephone company. Friends of mine at Raytheon in the Boston area have six portable units that they told me about in May of this year. I finally got delivery of a locally redesigned manufactured unit in October. I think General Motors had the prototype by this time and everybody wanted it. As an educator, I'd like to have the Dataport in my office, and when I have a bright idea, go to my classroom and show the students right then and there. In fact, I can put an opaque projector on it for 20 or 30 people. Better yet would be closed circuit TV in our classrooms. The cost of the terminal is really cheap in terms of a total outlay but is significantly expensive in terms of its relationship to other things. It is part of the inflexibility of the phone company and I think we ought to nurse them a little bit.