Computing as a tool for human augmentation

by W. J. Doherty
W. G. Pope

The IBM Thomas J. Watson Research Center in Yorktown, New York, has experienced a factor of twenty times increase in the past ten years in the amount of time its people spend using computers interactively in their work. This is twice the penetration rate of television in the 1950s. A similar degree of penetration is expected to happen in the rest of industry in the next ten years. That will mean a major departure from traditional data processing, with computers being used as tools to augment the users' abilities in all phases of their work. Examples of human augmentation, as seen in the work of several researchers, are offered in this paper. The integration of large numbers of personal workstations into this environment has given us new understanding of how to work. The causes, impediments, and consequences of these changes are described, with emphasis on human requirements for bandwidth, response time, tools, on-line storage, and computing capacity.

This paper discusses the use of computing as a tool to augment or enhance the memory and reasoning capabilities of staff members at the IBM Thomas J. Watson Research Center sites in northern Westchester County, New York. We use the familiar name "Watson Research" to designate the combined IBM Thomas J. Watson Research Center sites at Eastview, Hawthorne, Stormytown, and Yorktown.

Every tool augments or extends human capabilities in some way. The hammer extends the power of our fist; the automobile extends the stride of our legs; and the computer augments our memory and reasoning power. Our conclusion today is that the computing environment at Watson Research has significantly improved the memory and reasoning capabilities of many staff members by factors of 2 to 6 in quality and quantity for individual researchers during the past twelve years. We focus our attention here on the theme that people are much more expensive than computers for almost all uses of computers today, and the difference continues to grow. We discuss the architecture issues that must be addressed to make people more effective when using computers.

Intelligence is the combination of a person's abilities to learn to deal with the circumstances of living. Experience and knowledge contribute strongly to intelligence. We propose the metric *AQ*, or *Augmentation Quotient*, as an estimate of the actual *knowledge work* accomplished by a person with a given intelligence in a given time period, when aided by a computer, divided by the knowledge work accomplished in that same time period by the person alone. (Knowledge work is the mental equivalent of physical work.) The computer is a tool that, when properly used, can augment our intelligence considerably.

Areas of computing use

Computing has a long history, going back to the time of the pyramids and earlier, stimulated by human needs—to measure the land and its products and to construct the buildings on the land. In this paper, we present our thoughts on some of the needs and uses of modern electronic computing, which we believe is still in its earliest stages of development. As such, it is in an intermediate period that is common to major new inventions, when the new technology is mainly applied to reducing the cost of or speeding up existing ways of working. This kind of usage usually hides the primary value and utility of the new invention and can be expected to last for about two generations. Think how the automobile was regarded for many years as a horseless carriage. It took over forty years for modern highways, more reliable tires, and roomier and more comfortable cars to revolutionize transportation and our style of living along with it. Consider the telephone. The telephone was invented in 1876, and, for the next thirty years, it was marketed throughout Europe as a "broadcast" medium. In 1891, E. Bellamy wrote: "You stay at home and send your eyes and ears abroad for you. Wherever the electric connection is carried, it is possible ... for the dweller to take his choice of the public entertainments given that day in every city of the earth."[1] It was not until 1929 that the first telephone appeared in the Oval Office of the President of the United States.

Until recently, the primary uses of computing have been to automate existing ways of working.

Computing began as a commercially viable business about 1951. Until recently, the primary uses of computing have been to automate existing ways of working in banks, insurance companies, laboratories, and factories. Computerization of corporate processes such as payroll, order entry, supermarket checkout, inventory control, etc. are now commonplace in the world. A second major area of computer use lies in the real-time control of factories, oil pipelines, and automated assembly lines. These real-time control uses of computing are also widespread. A third major use of computing developed when microprocessors were embedded in games, calculators, cars, watches, ovens, refrigerators, and many other products.

The theme of this paper is that a fourth area of computing usage will be at least as important as those just mentioned above: computing used as a tool to augment human intelligence.

We doubt whether anyone can foresee the impact computers may one day have in our homes. There are certainly many, many other uses to which computers will be applied that have not been dreamed of as yet. Figure 1 shows a possible grouping of different areas in which computers have been applied.

Thomas Jefferson, negative intelligence, and AQ < 1

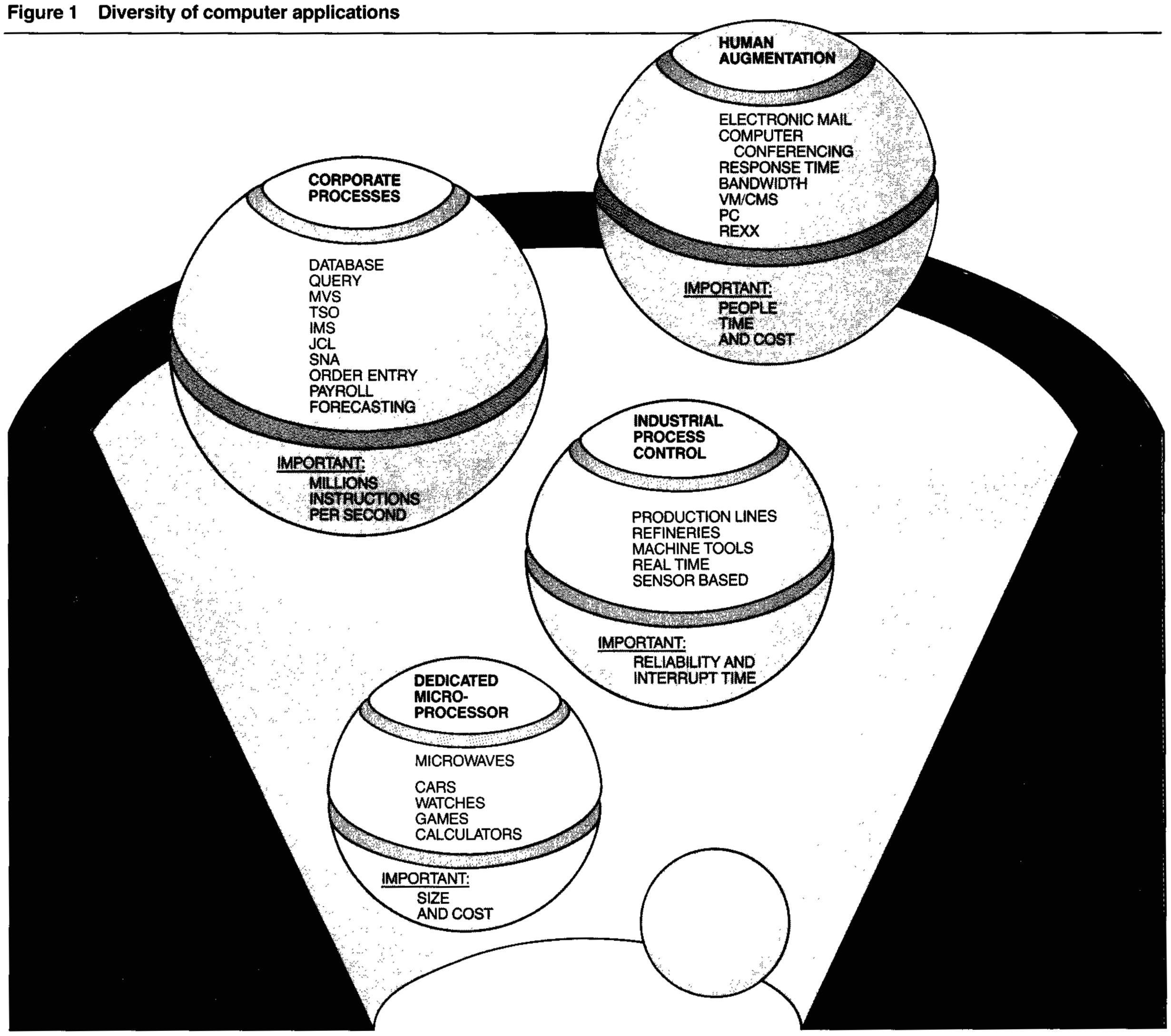
Many persons, including Thomas Jefferson, have noticed what might be called the intelligence of a group. Jefferson further postulated the notion of negative intelligence in the following way. Suppose a group of people had a certain group intelligence brought about by the synergistic interaction of its members. If a loud-mouthed dullard entered the group, Jefferson concluded that the group intelligence would decrease because more of the group's time would now be spent in dealing with ideas of lesser merit. The group's new member had introduced negative intelligence to the other members.

Similarly, the AQ of some computing environments can be less than one. This implies that people working in such an environment work less efficiently with a computer than they would without it.

Information pollution by the spread of false information, information dilution by excessive prompting of people from computer workstations, and excessive bureaucracy all lead to environments in which the AQ can be less than one. In some cases during the first thirty years in computing, people have automated functions before they understood them well enough to take that step, which has caused inconvenience and inefficiency for the users. In other cases, functions were automated too rigidly, which led to difficulty in adapting to changing requirements and rapidly growing volumes of data. Although the computer can be an intelligence amplifier when used properly, it can work against the user when improperly used. In the extreme, user–computer dialogues that had been programmed into applications led to users' taking two to three hours to do tasks that might have been doable in two to three minutes.

The automobile does not cause one to travel; it augments travel when necessary. Similarly, the computing environment does not cause one to be more productive; it allows one to be more productive when

Figure 1 Diversity of computer applications



the need arises. By studying the way scientists and engineers at Watson Research conduct their research activities, we can conclude that the computing environment at Watson Research has significantly enhanced the work done there. The first sign of improved productivity is improved clarity and quality of what is done. The second sign is improved quantity, which—in a research environment—is less important than improved quality.

The Watson Research computing environment

Figure 2 shows the computing environment at Watson Research as it was in April 1986. At that time, the number of staff members had outgrown the space available at the site, and the facility had spread into three other sites in Stormytown, Eastview, and Hawthorne, New York, all within a distance of seven to ten miles of the original site. Six IBM 3081s and an IBM 3090 provide the primary VM/CMS interactive services for these facilities. Ten IBM 4381s provide VM/CMS services for special projects.

There are about 3500 IBM PCs of many types and 3000 other terminals, mostly connected to the VM/CMS systems at data rates near 100000 bytes per second. VM/CMS on the mainframe operates as a high-speed local-area network (LAN) of virtual machines. We connect the workstations to each virtual machine and use the mainframe as a file server, print server, etc. to the workstation. There are over 800000000000 bytes of on-line, direct-access storage. Most of that is clustered so that it can be commonly accessed at each physical site. MVS/XA/TSO is run on an IBM 3084, a 3090 with a vector processor, and a 4381.

The Watson Research sites are connected together by T1 telephone lines. These are telephone lines that operate at a theoretical speed of 1.55 megabits per second. A speed of 1.45 megabits per second is actually achieved by a locally developed software extension to the PASSTHRU (PVM) product. This extension provides a packet-switched network for graphics, image, and high-speed file transfer, which means that a response-time delay of 0.1 seconds is the normal delay seen by a user who uses the mainframe via a T1 telephone line.

The IBM computer network, called VNET, connects more than 2000 systems at IBM sites in more than 225 cities around the world.

Over 1000 PCs are provided to IBM Watson Research staff members in their homes. A telephone line connection to the work environment is also provided at 1200 baud. This speed is improved by data-compression software developed as a research project that makes the line appear to the user as though it had 4800 baud capability. This is too slow for graphics or image or rapid scanning of files, but it does allow local editing, electronic mail, and some experiment control to be done from home.

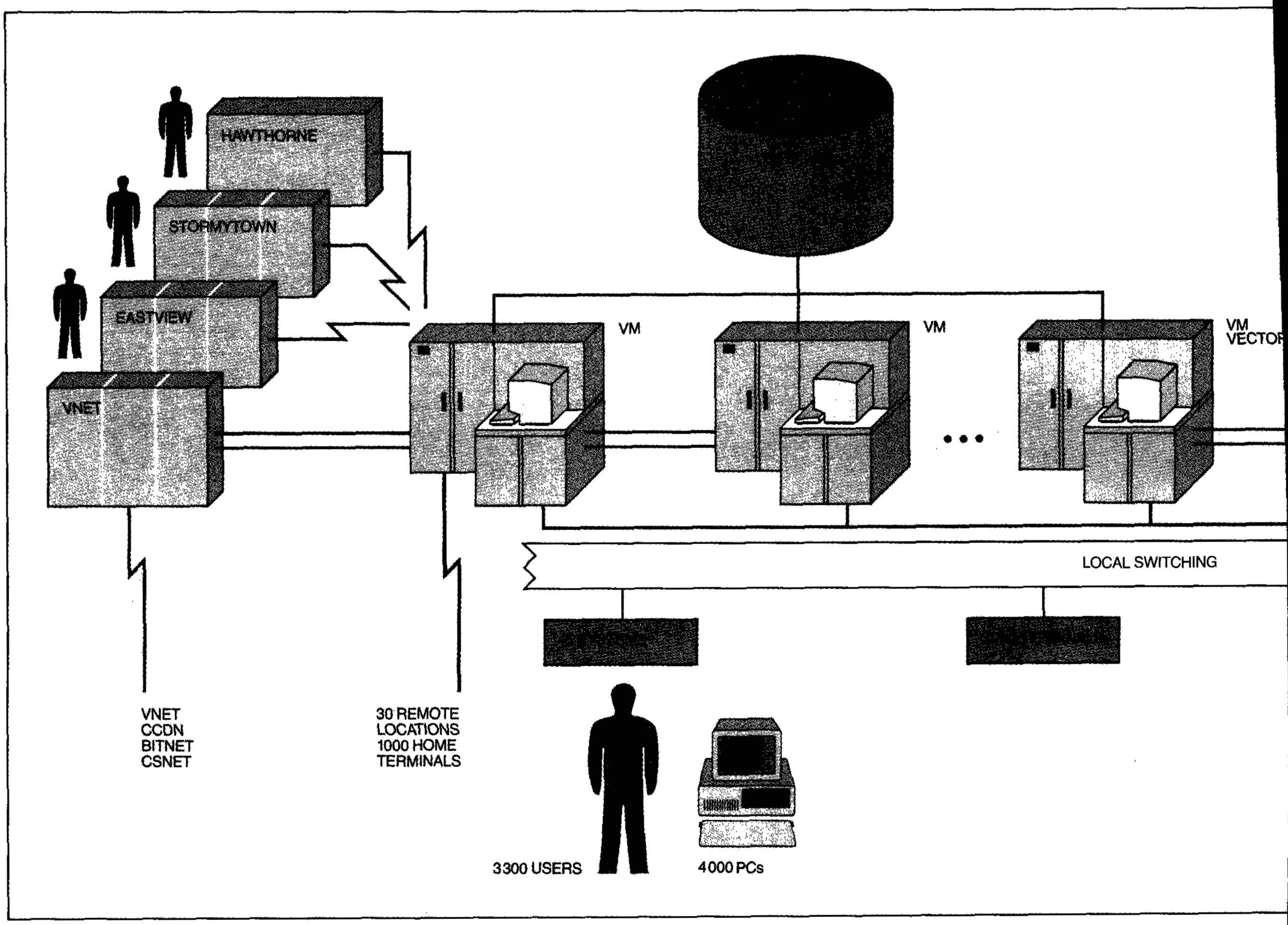
Graphics and text-processing services are provided with a broad spectrum of devices ranging from 200-picture-element- (pel) per-inch printers up to a 1400-pel-per-inch photocomposer. Image and graphics can be integrated with text to provide camera-ready copy for publication. Editing and text-processing tools, such as XEDIT (the System Product Editor), PE and SCRIPT, and the Generalized Markup Language (GML), provide powerful document creation and generation capabilities. An experimental editor for the IBM PC, called GEDIT, allows manuscript creation in a natural format and generates the GML tags for the user. An image editor, called IAX, developed experimentally at the IBM Winchester Scientific Center, has proved to be effective for the Watson Research staff.

The IBM computer network, called VNET, connects more than 2000 systems at IBM sites in more than 225 cities around the world. Conferencing systems using VNET enable new software tools to be developed and shared rapidly. For example, two of these conferencing systems deal with PC development and VM/CMS development. More than nineteen gigabytes per week of traffic are transmitted over this network from Watson Research. A network known as BITNET connects the Watson Research users to their colleagues in universities through the computing facilities of over 1500 systems in more than 400 sites around the world. A third network, called CSNET, connects Watson Research to computer science departments in universities around the world. These networks are each growing rapidly now. The growth rate at the start of 1986 was about two new systems per day.

The Watson Research computing systems staff strives to provide one hundred percent of its full range of services, twenty-four hours a day, seven days a week. The primary service objectives are response time of less than a third of a second for ninety percent of the interactions, as seen by the user, and an expansion factor (the ratio of elapsed time to do computation-intensive work, when others share the computer, to the stand-alone elapsed time) of less than 3.5 for ninety percent of the time. When these objectives are met, as they normally are, the Watson Research user has 3 to 16 millions of instructions per second (MIPS) of computing power, 4 to 8 megabytes of memory, large-capacity file systems, libraries of thousands of useful tools, and a worldwide network connection within IBM and among many universities.

Computation-intensive work is more cost-effective on a large host than on a workstation, by a ratio of about twenty to one at present. This ratio is expected to continue for the foreseeable future, because computation intensity is usually a function of the amount of data being processed. Thus large memories and fast CPUs are expected to continue to give the host

Figure 2 Computing environment for IBM Watson Research



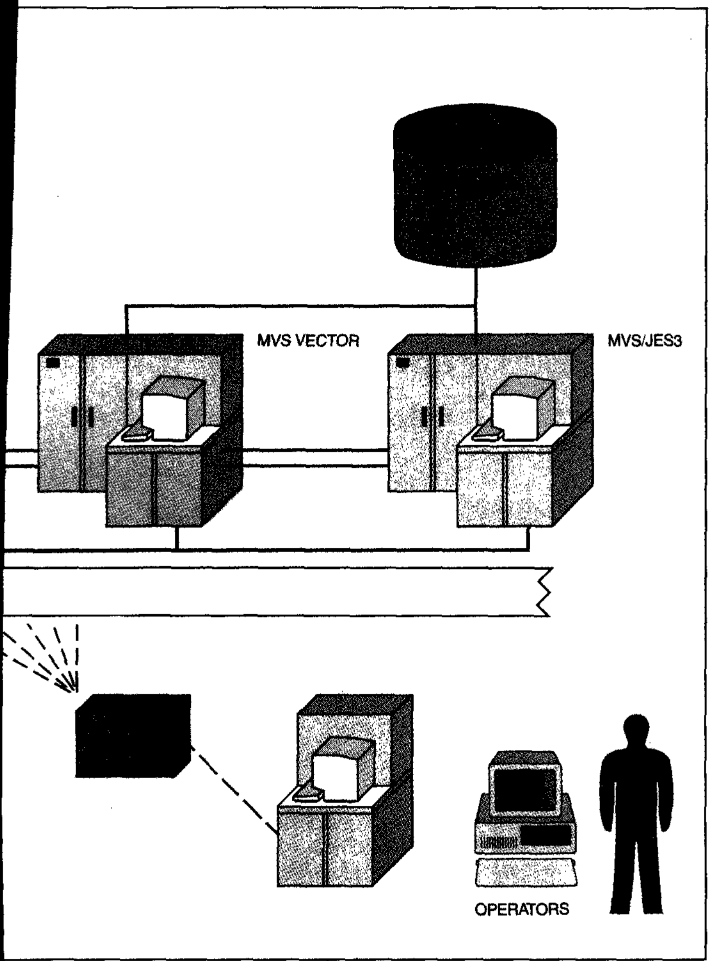
world an advantage in computation-intensive work for years to come.

Human-intensive work should be more cost-effective on a workstation due to the higher bandwidth and more rapid response available on a workstation.

Computing penetration at Watson Research

Computing libraries containing more than 3000 tools are available to the Watson Research staff. Systems normally come with about thirty tools. That initial system is like an acorn that grows into an enormous oak tree. The topic of interest is the oak tree, not the acorn. Thus we have carried out studies of the Watson Research systems and their users to understand better the evolving user requirements.

By offering the computer primarily as a tool to augment the memory and reasoning power of the staff members at Watson Research, we have achieved a major change in their ways of working. In 1979, we showed the growth of interactive computing at Watson Research from 1973 through 1977.[2] This growth has continued and represents a twentyfold increase since 1973 in the time spent using workstation display screens as effective windows into their work product. This increase is twice the growth of television in the 1950s, which was the period of greatest growth for television. Figure 3 shows the growth in interactive computing at Watson Research from 1973 through 1984. The aggregate value of the users' time, computed at a rate of $50 per hour, has grown to be much greater than the computer costs. This is shown in Figure 3, and this difference contin-



ues to grow. The data of Figure 3, quite conservative in 1984, are even more so today. We can calculate the value of an IBM employee's time today by dividing the worldwide gross annual income for 1985 of about $50 000 000 000 by an estimated 400 000 employees, which yields a current figure of $125 000 per person per year, or about $65 per hour today.

We have found that the distribution of computing time is far from even. Of the 3000 people who used the Watson Research computing systems in 1984, 80 individuals consumed 75 percent of the CPU time. We spent about $100 000 per person for each of those 80 people in 1984 to support this computation-intensive use. Although this is a large expenditure of money, it is less than the burdened yearly cost of a technical assistant.

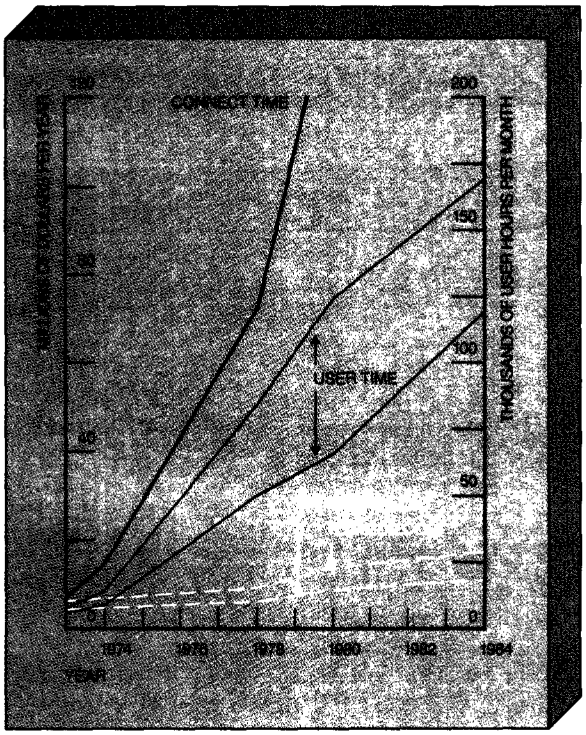
For the remaining 2920 people who used the Watson Research computing systems, we spent about $3500 per person in 1984 for human-intensive use. Our users generated about 3 000 000 interactions per day in the computing environment at Watson Research in 1984. If we could put the processing power of the world's most powerful supercomputer into each workstation, it is still doubtful that we could impact more than three percent of our people, unless we could improve the human–computer interface. This low degree of impact potential is because the limiting economic factors for 97 percent of our users of computing are the human–computer interface factors of function, bandwidth, response time, on-line documentation, and other aids to ease of use. Processing speed is the limiting factor for only three percent of our current users. As of year end 1984, the value of the active human time at the keyboard for the human-intensive work was 35 times greater than the cost of the computing service.

Augmentation through memory extension

A Watson Research mathematician, J. W. Cooley, was preparing to teach a class on Fourier transforms. To prepare the material for the class, he sent notes on our systems to his colleagues, asking for good examples that he could use in the class. Cooley's colleagues responded to his request in a few minutes with many examples representing real problems they had faced in their work and with correct solutions. Cooley read these examples into his file system, studied them, and found he had enough material for a substantial set of lectures. He then turned the examples into transparencies, using the Watson Research text-processing facilities and photocomposer. This whole process was completed in an hour.

Writing a program to use in his work twelve years ago took two weeks to two months. Today, it normally takes about half an hour to half a day. This is not because of any breakthrough in programming languages, but because of sufficient on-line disk storage to hold the cumulative work for the past twelve years of J. W. Cooley and his colleagues.[3,4] Today, more and more, persons doing research can extend what they have done before, combine functions they have previously programmed, or apply new limits and constraints to work they had done before. This increased capability means they can now spend far more of their time applying sound mathematical programs to solve more problems in better ways, more quickly than before. This facility gives them the opportunity to speculate and try many solutions to any one problem to produce one good solution.

Figure 3 IBM Watson Research computing growth in human and computer dollars



The automobile improves our ability to travel by a factor of twenty to one in speed compared to that of a healthy person walking, and thus makes possible travel in ways never previously imagined. The computer augments human memory and reasoning power in similar fashion, after we make the investment in human experience over time to fill up this on-line computer storage with meaningful human work in the form of programs and data. A new computer is like a newly born human being: It must be educated before it can become productive.

Augmentation through computation and graphics

A solid-state physicist, P. E. Seiden, spent two years working in the field of biophysics, using APL to build a potential evolutionary model. At the end of that time, H. Gerola, a visiting scientist, showed Seiden an article in the *Astrophysical Journal* by a third person about a theory of how spiral galaxies came about in our universe. Seiden found the galaxy theory of astrophysics remarkably similar to his own work in biophysics. He added two critical improvements to the astrophysical theory discussed in the article, spent two weeks converting all of his work from the past two years into polar coordinates, and then produced computer-generated models of spiral galaxies that conformed quite closely to actual galaxies.[5] Seiden has continued his work on spiral galaxies, which has been favorably reviewed in more than forty scientific publications around the world.

Growth of tools containing the stored experience of skilled users has led to reductions in the time required by the whole user community to do interactive work.

More recently, a scientist on a postdoctoral appointment from the Hale Observatory visited Watson Research and brought over one hundred photographic plates of galaxies. A barrier to further analyses of these plates, as with many others, was that the luminosity of some regions was so great that it masked the information that was there. The two scientists digitized one of the plates and entered the data into our system. Then they used an image editor called IAX, developed by P. H. Jackson in our Winchester Scientific Center, to remove the excessive luminosity. The resulting image was then displayed on a color display screen. From studying this display, the two scientists learned more about spiral structure in the galaxy in the first few days of this study than they had in the previous two years.

Augmentation through synergistic studies

A Watson Research mathematician, B. B. Mandelbrot, has developed a new branch of mathematics that deals with what Mandelbrot terms *fractals*. In the 1970s, Mandelbrot thought he had a good idea, but he had trouble convincing others of its merits. So he began writing computer programs to generate pictures of objects normally found in nature. At first this took about six months. Then several graphics programmers, S. Handelman, M. R. Laff, and V. A.

Norton, developed a better graphics interface for Mandelbrot and helped reduce the time to produce good fractal pictures to a few days. A solid-state physicist, R. F. Voss, improved upon their work, added to it, reduced the time still further, and then applied fractals to music.[6,7] Martin Gardner's Mathematical Games column in *Scientific American* in April 1978 contains a summary of fractals and their application to the description of music. B. B. Mandelbrot wrote several books about fractals, using the text-processing facilities of the computing environment at Watson Research.[8,9] Since then, fractals have been applied to describing computer chip designs, the United States economy, and many other areas. Movies such as "Star Trek" have fractal images in many scenes. Fractals have now spread to hundreds of new applications.

The networking capabilities of the IBM internal network, VNET, have facilitated the evolutionary growth of libraries of thousands of useful tools.

Augmentation using synergistic tools and computer conferencing

Today, several thousand programs called EXECs written in REXX, the VM System Product Interpreter, make up the primary user interface at Watson Research.[10,11] EXECs are the cumulative work product of tool builders in IBM locations all over the world. The average EXEC as invoked by a Watson Research user is a REXX program containing 157 commands. Twenty-seven of these commands would otherwise have had to be keyed in by the user. One hundred thirty commands are parameter-setting or flow-of-control commands that had been created by an experienced user for others who usually do not have those same skills to use. This growth of tools containing the stored experience of skilled users has led to significant reductions in the time required by the whole user community to do interactive work. Because of EXECs, each interaction between the user and the computer is now much more powerful than it was initially; the user accomplishes more work in less time.

The networking capabilities of the IBM internal network, VNET, have facilitated the evolutionary growth of libraries of thousands of useful tools. In 1977, P. G. Capek began to distribute an informal electronic newsletter describing useful tools that others had developed. This VM newsletter inspired the growth of internal conferencing systems devoted to tool development in different areas. There are now more than 70 different topics covered in the companywide conferencing systems. One primary topic today deals with tool development for VM/SP. A related internal conference deals with tool development for the IBM PC. Known as TOOLS, the work was carried out by M. F. Cowlishaw in the Winchester Scientific Center and D. M. Chess at Watson Research to run the conferencing systems.

The conferencing system dealing with the IBM PC now contains more than one thousand separate forums for discussion of the IBM PC and its software. This has the synergistic effect of focusing the attention of people all over the company on tool development in a timely fashion. New tools are developed faster and are put into use by our research and development people at a rate that is much faster than would otherwise be possible. By the start of 1986, one of every one hundred IBM employees had contributed to the PC conference, and about ten times that number had read items from it. Most of this worldwide community now benefits from using tools developed in this synergistic way.

Growth in updates (or *appends*, as they are also known) to the IBM PC conference are as follows:

Year	*Updates per week*
1982	10
1983	35
1984	210
1985	700
1986	2100

In the mid-1960s, computers were much more expensive than people. Thus, the time-sharing of computers developed as a way to distribute the expense of the computer. Now the computer in the form of a workstation costs much less than a person. Surveys taken of hundreds of engineering managers concerning the number of their engineers who could make effective use of workstations showed that only 5 to 15 percent of the engineers might be able to do so. If even these few engineers were left alone for a year each would no doubt evolve his environment in major ways, and almost all of those ways would very likely be incompatible with one another. The Tower of Babel would once again appear. Thus the expensive commodity of today's world is the user. To facilitate the effective use of a person's time, we need to create community environments of shared libraries. This encourages cooperative growth of tools and provides a common workbench for all to use. We are again integrating workstations with large mainframes to share the expensive resource of human knowledge.

Another example of synergism that is manifesting itself was shown when a salesperson in another country was in a competitive situation for a major computer installation at a university. The salesman needed information on how to interface IBM computers with those of another manufacturer. His query through VNET for help brought him hundreds of replies totaling 500 pages of information that led to winning the bid.

Our final example of work produced synergistically is that of the REXX language itself.[10,11] REXX, a cross between PASCAL and PL/I, was developed by M. F. Cowlishaw, with constructive suggestions from about a dozen other experts scattered throughout the IBM organization. It went through more than 30 versions before achieving widespread use between 1979 and 1982. Since 1982, it has spread to the point where it is now the most widely used programming language in IBM. It received wide acclaim from customers after its introduction as a product in 1983.

Needs for high bandwidth and subsecond response

As a result of asking thousands of users how they spend their time working with information, we find that almost all of them estimate that they read only one tenth of one percent to five percent of the information they scan, and the percentage of time spent writing is very small compared to the time spent reading. Thus, if we want to apply the computer to more of the information-handling tasks people do, we must cater more to scanning and reading. This requirement becomes especially significant once people have accumulated large amounts of information on line.

Scanning large volumes of information quickly requires a much higher bandwidth from computer to display than is usually provided for alphanumeric work. Graphics and image display needs raise the bandwidth requirements to the range of 1 million to 100 million bits per second. Typing skill requirements diminish as more and more information becomes available on line. Most interactions involve one or two keystrokes in an editor to move around information that is already there.

In dollars, one computing environment may be as much as 25 times more cost-effective than another.

During the first 30 years of electronic data processing there was very little disk storage, and it was much more expensive than today. Therefore, users entered their data from a keyboard more frequently than they extracted it from an on-line disk. Because most users were in this stage of computing in the early days (which was also known as the bring-your-own-bits or BYOB stage), emphasis was placed on clever ways of inputting and anticipation of what the user wanted to type via full duplex ASCII terminals. Today, the very low bandwidth and slow response of these older terminals acts as an impediment to scanning information or to doing graphics, image processing, or animation on the screen.

Many studies have now been completed to show the importance of subsecond response time as an essential requirement for human productivity via computing.[2,12–14] To illustrate the economic importance of human productivity, we carried out a normative study of cost per interaction in December 1980 for seven systems within IBM. The results show a range of 25 to 1 in cost per interaction, when user time at the keyboard is considered as part of the cost. Thus the AQ a person gets from one computing environment may be as much as 25 times greater than that

Table 1 Real cost of an interaction for two different types of systems—A and B

Measure	System				
	A	B1	B2	B3	B4
Number of logged users active in 10-minute increments	300	245	70	100	25
Total interactions per year	237600000	76320000	34560000	23040000	14400000
Interaction rate per system	33/s	10.6/s	4.8/s	3.2/s	2.0/s
Interaction rate per person	396/h	156/h	250/h	115/h	288/h
Internal response time (s)	0.4	1 to 12+	0.84	2.2+	0.9
Terminal location	LOCAL	REMOTE	LOCAL	REMOTE	LOCAL
Interactive I/S cost	$3195300	$7900000	$2064500	$1287000	$886500
Lost work	$225000	$11200000	$1968750	$5343250	$525000
Computing cost per interaction	$0.013	$0.10	$0.06	$0.056	$0.062
Total cost per interaction	$0.014	$0.25	$0.117	$0.288	$0.098

received from a different computing environment. In dollars, one computing environment may be as much as 25 times more cost-effective than another.

Table 1 contains data that approximate the cost per interaction of two different computing environments; that is, the data represent the cost of the human-intensive interactions, which dominate the total man–computer interactions. The computing resources consumed by human-intensive work were less than 35 percent of the total Information System (I/S) cost. The basis for comparison is a 250-day year having eight hours per day of interactive work.

The line called "Lost work" is based on studies of the human–computer interface. For the human-intensive work, the computing resources consumed per interaction tend to be quite small, about 10 ms of 3081 CPU time, ten paging I/O operations, and one or two disk I/O operations. A. J. Thadhani has shown that people naturally gravitate toward working at more than 400 interactions per person per hour when the computer is not an inhibiting factor.[12] Other evidence suggests at least 600 interactions per person per hour and engineers using interactive graphics approach 4000 interactions per person per hour.[14]

The value of the person's time for human-intensive work in 1980 was more than 20 times greater than the cost of the computing. Therefore, it is important to configure systems so that the system user is not inhibited by the computer. Thadhani has shown that people slow down by a factor of 4 in their rate of working as the internally measured system response time with locally attached 3277s moves from 0.5 seconds to 2.0 seconds for remotely attached 3277s.[12]

The average value or productivity of a person's time in IBM in 1980 was $75000. The line called "Lost work" is based on the assumption that 400 interactions per hour is a minimal good working rate for people. It is probably greater than 400, but the assumption of 400 shows strong enough results. "Lost work" is calculated as follows:

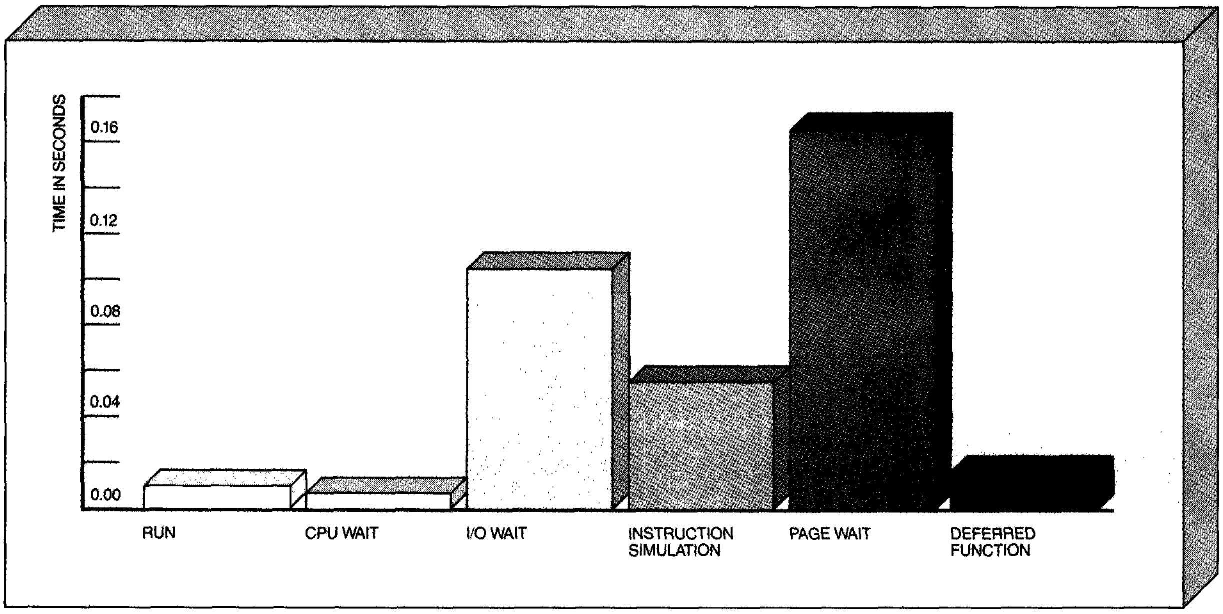
$$\$75000 \times [(400 - \text{interactions per hour})/400] \times \text{number of logged users}$$

The "number of logged users" is an estimate of the users who are both logged on and active over ten-minute intervals of time.

System management controls

Attaining excellent response time and availability requires timely management and long-range planning. Keeping response time within specifications is important at Watson Research because even slight degradation in interactive service can have a major impact on users' continuity of thought. Thus, with our dominant theme of computing as a tool for human augmentation, the process begins with a robust capacity-planning procedure that avoids unnecessary complication. At Watson Research, capacity

Figure 4 Components of response time, January 1983



is defined in CPU minutes per day, a measure that the user community can understand and associate with their activities. The machine plan is defined by matching the demand from the user community with the supply capability of the computers installed. *Demand* is estimated by combining historical trends in usage with projections of new, large projects planned to start within the next six to eighteen months. *Supply* is established from historical records of CPU minutes delivered versus service delivered. Dividing demand by supply provides an estimate of the capacity that would be needed to fully meet user needs. This number is then adjusted by the management process to meet division budgetary restrictions.

Once the machine plan is accepted, the *available supply* is established. A process of *allocation* begins, and the supply is proportioned among the using departments, each allotted a share established by relative demand. The key to making the allocation process work is a regulatory mechanism running on the Watson Research VM computers, called NOW, that constantly monitors service levels. As long as the service levels are being met, NOW takes no action to affect user priorities. However, if the service levels are exceeded, NOW acts to reduce the rate of CPU consumption by users in departments that have exceeded their allocation. This action affects primarily those users who are doing computer-intensive work at that time and quickly restores performance to the specified levels. When performance has remained within the specified levels, NOW relaxes the constraints until all users are again running at the standard priority and performance is uniformly acceptable. These two processes of robust capacity planning and automated service controls are the foundation for achieving the service levels specified at the Watson Research site.

Service levels at Watson Research are under continuous investigation for improvement, because the better the response time, the better the user productivity. Day-to-day performance management activities address unique performance problems. Sustained performance improvements are identified only by careful analysis of usage data.[15] The technique of state sampling the service received by each user, on a continuing basis every day for every user, allows us to determine which system resource might be a present or future bottleneck to good service.

Figures 4, 5, and 6 show the changes in the profile of where an average Watson Research user's VM/CMS interaction spent its time from January 1983 through September 1984. The vertical axis is time-delay-measured in hundredths of a second. The diagrams illustrate that for most transactions CPU time is not the major bottleneck, because the transaction spends

Figure 5 Components of response time, November 1983

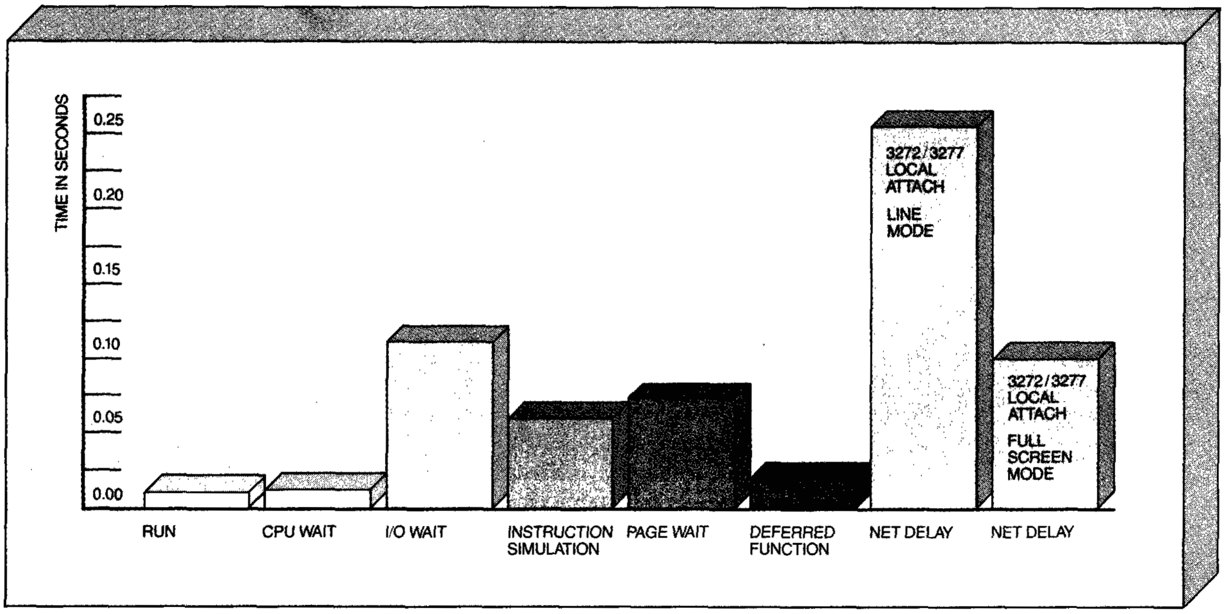
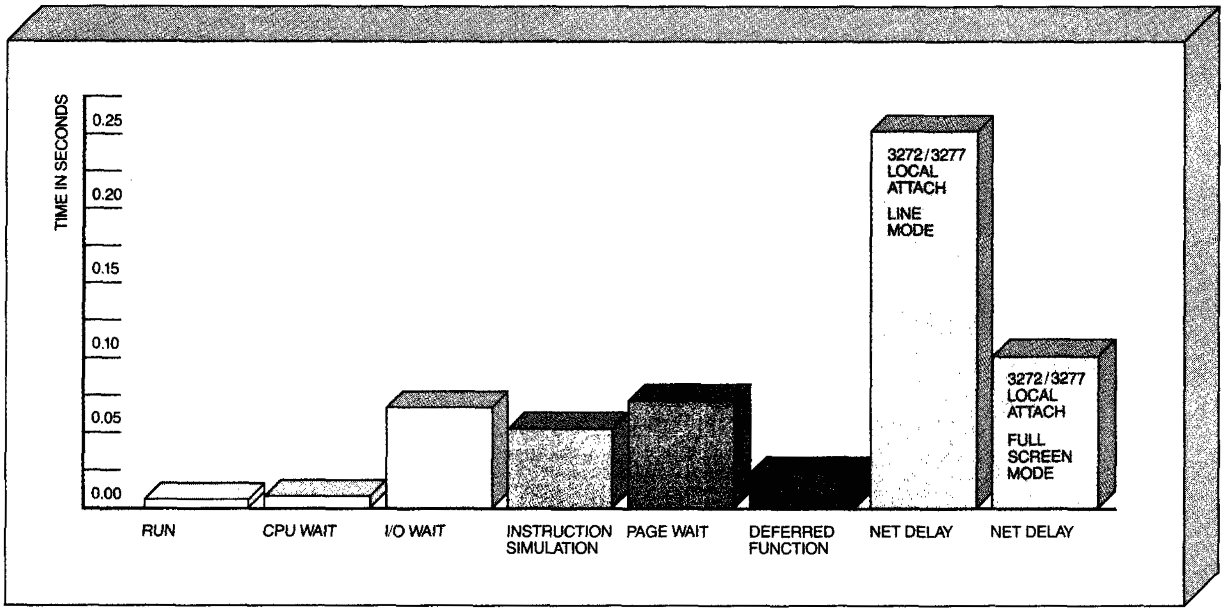


Figure 6 Components of response time, September 1984



much more time waiting for paging and I/O to complete. The installation over the past two years of solid-state paging and disk-cache devices has allowed response time to the 90th percentile to drop from 0.55 seconds to 0.35 seconds. Current analysis shows that the time delay of a user's interaction in the terminal control unit now overshadows the host computer delays. This has caused Watson Research technical teams to search for superior technology to connect users with community computing services at ever-increasing bandwidths.

Cooperative processing between workstations and hosts

Workstations connected to community systems in a cooperative fashion are expected to increase the bandwidth to the user and achieve still lower response times (to about 0.1 seconds).

Studies of the usage of interactive computing indicate that 44 percent is associated with editing and text processing, and another 21 percent with electronic mail.[16] Therefore, 65 percent of a user's time is associated with text-manipulating tasks. Another 15 percent is related to file management activities that might be considered rearranging one's electronic desk. That leaves a residue of only 20 percent of a user's time associated with such computer-intensive activities as programming, data analysis, graphics, and other such tasks.

A 1984 study of 45 Watson Research users showed that the introduction of personal computers (PCs) had done little to change the typical Watson Research user's computing habits.[17] Most PCs were used as displays that could also function as personal computers. The use of computing could be tied to a user's work activities, which have not changed much in the past ten years. Most changes in user behavior that could be attributed to PCs are ones of personal taste on the user's part.

Cooperative processing can be likened to the situation of commuting to work. A person who lives in the suburbs drives a car to the train station and takes a train into a major city to work. Railroad trains cannot match the personal convenience of the auto, nor can an auto match the passenger-carrying efficiency of a train.

The analogy applies to cooperative processing, which is a blend of the relative advantages of different levels of computing. The workstation, like the car, is personal and highly available and gives immediate response. Bit-map displays will allow for color, graphics, image, and eventually motion at high bandwidth. One can personalize the viewing screen like arranging work on a desk. Workstations give immediate feedback to every keystroke or the movement of an analog pointing device such as a mouse. Community processors can better manage tasks such as those requiring shared program libraries, large memories, central data bases, computing conferencing facilities, high-speed calculations, network connections, and batch operations. Community processors, properly managed, continue to perform the computer-intensive tasks more efficiently, and workstations serve the human being's intensive-computing needs.

The steps to cooperative processing are high-speed connection, communication, and adaptation.

The steps to cooperative processing are high-speed connection, communication, and adaptation. We can connect PCs to VM/CMS and MVS/TSO as display terminals. The PC functions either as a personal computer or as a host display, and the two interact primarily by transferring data back and forth between the host and the PC. Communication comes when programs in the PC can work in parallel with programs at the host, sharing real-time data and flow of control. Thus the two computers can work together on the same task with each computer doing the part of the task it does best, at a speed that is fast enough that any delay is undetectable by the user. Adaptation is built upon communication, as applications are built to solve problems for individuals. This is the direction in which our development activities are rapidly moving, and the synergistic conferencing environment previously described is the primary catalyst to accelerate this growth.

Concluding remarks

Computing is still in its infancy, a developmental state in which a culture gap separates those who understand the true value of computing from those who do not. Side effects arising from the misuse of computing have caused people to take longer to do some tasks with the aid of computers than they otherwise would. The use of computers as tools to augment human intelligence should enable us to reduce the ineffective use of computers while simultaneously improving our abilities as knowledge workers.

AQs of 2 to 6 are experienced by many scientists at Watson Research in our rapidly expanding computing environment. Where computation-intensive work is done, the AQ may be even greater than 100. This quantification is itself embryonic in nature and subject to speculation. It is the result of much reflection on the subject and should provide the basis for

The challenge is to make the human–computer interface simple and adaptive to the user's needs.

refinements to come. Personal gains come mainly from the ability of people to keep their work in a live form on disk storage for long time periods. The ability to reuse past work has also proved to be valuable. Secondary gains come from the ability to use the work of others. Synergistic gains occur from shared libraries, conferencing, and the ability of many tool builders to work together though physically far apart. The synergistic gains benefit everyone. The REXX language seems to be one of the best mechanisms we know of at the present time to make our systems more effective and easier to use.

Rapidly dropping computer hardware costs have reached the point at which hardware speed rather than cost is a limiting economic factor for less than three percent of our use of computers. The human time at the keyboard has a value that is 35 times greater than the hardware consumed for 97 percent of our use of computers. Therefore, we focused our attention on the requirements that must be met for people to make themselves more effective through the use of computers.

We found that bandwidths ranging from 1 megabit per second up to 100 megabits per second are required between the display window and the user's information, wherever it is stored. We found that subsecond response time, probably down to 0.1 second, is important to avoid disrupting the user's rate of work.

Cooperative processing, with a smooth blending of workstation and host capabilities, appears to be the solution for opening the bandwidth to the user further than has been achieved thus far. People change their work habits slowly and only when they perceive a personal benefit for their own work. The challenge is to make the human–computer interface so simple and adaptive to the user's needs that the benefit is immediately obvious to most users.

Acknowledgments

The authors acknowledge the contributions of all members of the Computing Systems department at Watson Research for continuing to provide the best interactive computing service that they have ever experienced. The authors give special mention to the following persons for their direct impact on this work. O. G. Selfridge first influenced us to focus on human augmentation. D. C. Engelbart directed research on augmentation of the human intellect at the Stanford Research Institute from 1957 until 1977. P. H. Callaway, W. H. Tetzlaff, D. N. Smith, D. H. Potter, A. B. DeMaio, A. M. Gottlieb, and M. H. Wickham built the tools that were instrumental in our analysis. S. J. Boies and A. J. Thadhani provided keen insight on response time requirements. M. B. Rosson provided useful analysis of editor usage. H. Serenson, R. P. Kelisky, and A. H. Weis provided the management encouragement for this work.

Cited references

1. A. Briggs, "The pleasure telephone," *The Social Impact of the Telephone*, de Sola Pool, Editor, The MIT Press, Cambridge, MA (1976), p. 40.
2. W. J. Doherty and R. P. Kelisky, "Managing VM/CMS systems for user effectiveness," *IBM Systems Journal* **18**, No. 1, 143–163 (1979).
3. R. C. Agarwal, J. W. Cooley, F. G. Gustavson, J. B. Shearer, G. Slishman, and B. Tuckerman, "New scalar and vector elementary functions for the IBM System/370," *IBM Journal of Research and Development* **30**, No. 2, 126–144 (1986).
4. R. C. Agarwal and J. W. Cooley, "Fourier transform and convolution subroutines for the IBM 3090 Vector Facility," *IBM Journal of Research and Development* **30**, No. 2, 145–162 (1986).
5. P. E. Seiden and H. Gerola, "Propagating star formations and the structure and evolution of galaxies," *Fundamentals of Cosmic Physics* **7**, 241–311 (1982).

6. R. F. Voss, "Random fractal forgeries: From mountains to music," *Science and Uncertainty*, S. Nash, Editor, IBM UK Ltd., Science Reviews Ltd., Middlesex, England (1985), pp. 69–86.
7. M. Gardner, "Mathematical games," *Scientific American* **238,** No. 4, 16–32 (1978).
8. B. B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman and Company, San Francisco, CA (1982).
9. B. B. Mandelbrot, *FRACTALS, Form, Chance, and Dimension*, W. H. Freeman and Company, San Francisco, CA (1977).
10. M. F. Cowlishaw, *The REXX Language: A Practical Approach to Programming*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1985).
11. M. F. Cowlishaw, "The design of the REXX language," *IBM Systems Journal* **23,** No. 4, 326–335 (1984).
12. A. J. Thadhani, "Interactive user productivity," *IBM Systems Journal* **20,** No. 4, 407–423 (1981).
13. G. N. Lambert, "A comparative study of system response time on program developer productivity," *IBM Systems Journal* **23,** No. 1, 36–43 (1984).
14. *The Economic Value of Rapid Response Time*, GE20-0752, IBM Corporation (1982); available through IBM branch offices.
15. W. H. Tetzlaff, "State sampling of interactive VM/370 users," *IBM Systems Journal* **18,** No. 1, 164–180 (1979).
16. B. Pope, *A Study of How Users of Programmable Work Stations Can Affect VM/CMS Usage*, Research Report RC-10721, IBM Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 (1984).
17. D. C. Engelbart, "The augmented knowledge workshop," *Proceedings, ACM Conference on the History of Personal Workstations* (1986), pp. 73–83.

Walter J. Doherty *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598.* Mr. Doherty is currently Technical Consultant for Computing Systems in the Computing Systems Department. He has been an Adjunct Faculty Member at the IBM Systems Research Institute, a Distinguished Visitor for the IEEE Computer Society, and a National Lecturer for the ACM. Mr. Doherty has received Outstanding Contribution awards for the design of IBM's first computer-assisted instruction system product, for developing scheduling strategies for IBM's interactive computing systems, for technology transfer, and for his response-time studies. He also received the Data Processing Division President's Merit Award for contributions to interactive computing products. Mr. Doherty has developed instrumentation and performance-measurement technology and used that instrumentation to study the man–machine interface and human productivity resulting from the use of computers for the past twenty years. He joined the IBM Research Division in 1960 and spent his first three years working on the development of the Master Control Program (MCP) for STRETCH. He has a B.S. in mathematics from Manhattan College and an M.S. in mathematics from New York University's Courant Institute.

William G. Pope *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598.* Mr. Pope joined IBM in 1966 as a systems programmer at the Endicott manufacturing plant, where he worked on the Common Manufacturing Information System (CMIS). In 1974, Bucky, as he prefers to be known, became a systems architect for CMIS. In 1976, he began to investigate advanced applications development techniques and to lecture on this topic. During 1978, he moved to the IBM Information Systems and Technology Group in White Plains, where he participated in a research project on enterprise analysis. In 1979, Mr. Pope became a program manager for application development productivity for the same group. He participated in projects on the applications development methodology and helped develop better quality techniques for Information Systems organizations. In 1983, he joined the IBM Research Division to study the value of computing to human beings. Since then, he has written three papers and presented them at major computing conferences. He is now the manager of Systems Management Controls and User Productivity at Watson Research, where he continues his work on making computers a valuable tool for people. Mr. Pope holds B.S. and M.S. degrees in business from the State University of New York at Binghamton.

Reprint Order No. G321-5277.