

THE AUGMENTED KNOWLEDGE WORKSHO

From *A History of Personal Workstations*,
Adele Goldberg [Editor], ACM Press, NY,
1988, proceedings of the ACM conference in
Palo Alto, CA, Jan 9-10, 1986 (101931).

Doug Engelbart
Introduction by Charles Irby

When I left Glen's group, I spent a year working for Lytton Industries on the ground control system supporting a precursor to skylab. Then I went to a conference that was being held up in San Francisco. It was the Fall Joint Computer Conference. This particular conference was very, very special for me and for a lot of people because there was an hour and a half session in which Douglas Engelbart from the Augmentation Research Center at SRI in Menlo Park, California, gave a live video demonstration of a highly interactive computer graphics and computer text manipulation system that had been developed at SRI. I went to that particular session not knowing what to expect, and I was completely blown away. I happened to find afterward the particular person who seemed to be in charge technically; his name was Bill English. I cornered him and said, "This is really nifty and I think I can help you." And he said, "We're looking for a few good men. Why don't you come by?" The next week I went by and the personnel department said, "No, we have no openings." I said, "Wrong. I'm going to sit here until Bill English comes to talk to me." (I was fairly bold in those days.) And he did. They hired me, and it was the beginning of a seven-year effort on my part at SRI with Doug Engelbart. A great many of the concepts that people now think of as commonplace, like the mouse and multiple windows, editing across windows, integrated text and graphics—all those things originated, from my point of view, from Doug and his group at SRI.

I'd like to introduce Doug Engelbart and his perspective of the last 20 years or so.



Doug Engelbart is a pioneer in the design of modern interactive computer environments. He holds the patent on the mouse; created the first two-dimensional editing system, and was the first to demonstrate such things as the use of remote procedure protocols, mixed text-graphics, and shared screen viewing.

Doug holds a B.S. in EE from Oregon State University and a Ph.D. in Electrical Engineering from the University of California at Berkeley. His career has led him from the U.S. Navy as an electronics technician to NASA's Ames Research Laboratory and then to Stanford Research Institute (now SRI International) where he led the team that designed and built the NLS "Augmented Knowledge Workshop."

From January 1977 to March 1984 Doug was a Senior Scientist at Tymshare, Inc., Cupertino, California. In 1984, Tymshare was acquired by McDonnell Douglas. Doug continues as a Senior Scientist in the Information Systems Group, promoting the type of integrated-system architecture conceived and implemented by him at SRI International.

The Augmented Knowledge Workshop

Doug Engelbart
McDonnell Douglas

Introduction

The story of my involvement with on-line workstations begins in early 1951 with a vision and a life-time professional commitment. Over 34 years of pursuit have created a lot of personal history, and the object of this historical exercise, the workstation, occupies a unique place in it.

For me, a workstation is the portal into a person's "augmented knowledge workshop"—the place in which he finds the data and tools with which he does his knowledge work, and through which he collaborates with similarly equipped workers. I consider that the large system of concepts, skills, knowledge, and methods on the human side of the workstation has to be taken into account, in a balanced way, when pursuing increased human effectiveness. So, my workstation-history story embraces a rather large sphere.

The task of writing an historical piece is unfamiliar enough to cause me difficulty by itself, but the associated stirring of old records and old memories has added a nearly overwhelming burden of dreams, events, people, stresses, pleasures, disappointments—the firsts and the failures. Now, what from all of this—and how to organize it—will make an appropriate "history" paper? I could provide a solid measure of objective reporting—events and dates. I have been an involved observer of related computer history since 1951. I watched and experienced how the supportive hardware, languages, and architecture evolved; witnessed the people and efforts that brought time-sharing into being; and was even more closely involved with the emergence of computer networks. Through all of this, I was wholly focused on what these things could do for people at workstations. And then there was office automation and personal computers; you don't have to be an old guy to have watched these emerge, but I'm sure they looked different to me than to most.

I could also provide lots of objective reporting about the events and dates associated with the things I have caused or had a direct hand in. There seems to be a lot there that is quite relevant to this "history of the workstation" theme. It was dusty, laborious work, this process of brainstorming for candidates, culling and ordering and trying to describe them in some reasonable sequence and context. But what I came

to realize is that there is one, clearly dominant factor that underlies essentially every cause for any uniqueness that I might list for historical record. It isn't a technology, it isn't a science, and it isn't a marketing or business model. And I am going to give it dominant coverage in this paper. It is what I call my "Framework." My Framework is based upon an intuitive conviction, implanted in my head (apparently permanently) over 30 years ago, that the gains in human knowledge-work capability that we will achieve by properly harnessing this new technology will be very large. Metaphorically, I see the augmented organization or institution of the future as changing, not as an organism merely to be a bigger and faster snail, but to achieve such new levels of sensory capability, speed, power, and coordination as to become a new species—a cat.

Based upon this conviction about huge potential gains for mankind, my Framework explains for me generally where such gains are going to come from, and provides strategic principles that can help guide a conscious pursuit of these gains.

Genesis

I was several years out of school, possessing a B.S. in electrical engineering and two years' experience during World War II (halfway through college) as an electronics technician. I was doing odd-job electrical engineering work at Ames Research Laboratory in Mountain View, California, with the National Advisory Committee for Aeronautics (NACA, forerunner of NASA). For several months I had been devoting most of my spare time to searching for professional goals; for some reason I wanted to invest the rest of my heretofore aimless career toward making the most difference in improving the lot of the human race.

I had initially dashed off in many fanciful directions, but yet managed enough interludes of reasonably sober thinking to build up some useful, strategic generalizations. Retreading myself professionally, to become proficient and then extraordinarily productive in some new field wasn't worth considering without a significantly attractive scenario, embedded in a reasonably structured strategic framework. The high-payoff scenarios all seemed to involve creating or joining something that, however disguised, would essentially be a crusade. Crusades have many strikes against them at the outset. In particular, they don't connect to a normal source of government or business revenue. They don't have nice organizational frameworks. You can't go out on the streets and expect to find financial, production, or marketing vice-presidents interested in the crusade. Moreover, even if you accomplish the sweeping change that is the ultimate objective, chances are that in

this very complex world the side effects might be bad enough to make you wish you hadn't tried.

Suddenly, up through all of this delightful, youthful abstraction bobbed the following clear realization: The complexity of the human situation was steadily increasing; not only that, but its rate of increase was increasing. Along with the increasing complexity had come a general increase in the urgency associated with the more critical problems. If one invented a measure for each of these—complexity and urgency—then for a given problem, the product of its complexity times its urgency would represent a fair measure of the difficulty mankind would find in dealing with that problem.

- FLASH-1: The difficulty of mankind's problems was increasing at a greater rate than our ability to cope. (We are in trouble.)
- FLASH-2: Boosting mankind's ability to deal with complex, urgent problems would be an attractive candidate as an arena in which a young person might try to "make the most difference." (Yes, but there's that question of what does the young electrical engineer do about it? Retread for a role as educator, research psychologist, legislator, . . . ? Is there any handle there that an electrical engineer could . . . ?)
- FLASH-3: Ahah—graphic vision surges forth of me sitting at a large CRT console, working in ways that are rapidly evolving in front of my eyes (beginning from memories of the radar-screen consoles I used to service).

The imagery of FLASH-3 evolved within a few days to a general information environment where the basic concept was a document that would include mixed text and graphic portrayals on the CRT. The imagery carried on to extensions of the symbology and methodology that we humans could employ to do our heavy thinking. There were also images of other people at consoles attached to the same computer complex, simultaneously working in a collaboration mode that would be much closer and more effective than we had ever been able to accomplish.

Within weeks I had committed my career to "augmenting the human intellect." In a few months, I left the NACA and enrolled as a graduate student at UC Berkeley, where Professor Paul Morton had started a computer science activity (although it would be many years before universities began calling it that). He was several years along in developing the California Digital Computer (CALDIC).

Within a few years I had to accept the fact that research on any kind of interactive computer applications would not provide me with a program acceptable to the university community for Ph.D. and later

faculty pursuit. So, I settled for something else, got my Ph.D., and went to Stanford Research Institute (SRI) where I hoped ultimately to promote support for an augmentation program.

FRAMEWORK

That was 1957. By 1959 I was lucky enough to get a small grant from the Air Force Office of Scientific Research (AFOSR, from Harold Wooster and Rowena Swanson) that carried me for several years—not enough for my full-time work, but by 1960 SRI began pitching in the difference.

It was remarkably slow and sweaty work. I first tried to find close relevance within established disciplines. For a while I thought that the emergent AI field might provide me with an overlap of mutual interest. But in each case I found that the people I would talk with would immediately translate my admittedly strange (for the times) statements of purpose and possibility into their own discipline's framework. When rephrased and discussed from those other perceptions, the "augmentation" pictures were remarkably pallid and limited compared to the images that were driving me.

For example, I gave a paper in 1960 at the annual meeting of the American Documentation Institute, outlining the probable effects of future personal-support use of computers. I discussed how a focus on personal support would change the role of their future systems and how such a change would enable more effective roles for the documentation and information specialists.¹ I received no response at all at the meeting. One reviewer gave a very ho-hum description of the paper as the discussion of a (yet another) personal retrieval system. Later, during a visit to a high-caliber research outfit, an information-retrieval researcher got very hot under the collar because I wouldn't accept his perception that all that the personal-use augmentation support I was projecting amounted to, pure and simple, was a matter of information retrieval—and why didn't I just join their forefront problem pursuits and stop setting myself apart?

Then I discovered a great little RAND report written by Kennedy and Putt² that described my situation marvelously and recommended a solution. Their thesis was that when launching a project of inter- or new-discipline nature, the researcher would encounter consistent problems in approaching people in established disciplines. They wouldn't perceive your formulations and goals as relevant, and they would become disputative on the apparent basis that your positions were contrary to "accepted" knowledge or methods. The trouble, said these authors, was that each established discipline has its own "conceptual framework." The enculturation of young professionals with their discipline's framework begins in their first year of professional

school. Without such a framework, tailored for the goals, values, and general environment of the respective discipline, there could be no effective, collaborative work. Furthermore, if such a conceptual framework did not already exist for a new type of research, then before effective research should be attempted, an appropriate, unique framework needs to be created. They called this framework-creation process the "Search Phase."

So, I realized that I had to develop an appropriate conceptual framework for the augmentation pursuit that I was hooked on. That search phase was not only very sweaty, but very lonely. In 1962, I published an SRI report entitled, "Augmenting Human Intellect: A Conceptual Framework."³ With the considerable help of Rowena Swanson, this was condensed into a chapter of a book published in 1963.⁴ I can appreciate that these framework documents appear to many others as unusably general and vague. But, for me, the concepts, principles, and strategies embodied in that framework look better and better every year. The genesis of most of what was and is unique about the products of the augmentation work can be traced back to this framework, which I'll return to later with a fuller description.

PROGRAM SUPPORT

I submitted many proposals before getting support to pursue the augmentation program outlined in the Framework report. Among the stream of politely phrased regrets, there was one that, in contrast to today's environment, can provide useful perspective on the environment of 1961. Four high-quality civilian experts had been enlisted by one agency as a site-visit team—a brain researcher, a psychologist, and a computer expert—and for me it was a very enjoyable day's dialog. But the subsequent letter from the agency informed me regretfully that (paraphrased) ". . . since your interesting research would require exceptionally advanced programming support, and since your Palo Alto area is so far from the centers of computer expertise, we don't think that you could staff your project adequately. . . ."

When J. C. R. Licklider ("Lick") came from Cambridge to take over ARPA's newly formed Information Processing Techniques Office (IPTO) in late 1962, I was figuratively standing at the door with the "Conceptual Framework" report and a proposal. There the unlucky fellow was, having advertised that "man-computer symbiosis," computer time-sharing, and man-computer interfaces were the new directions. How could he in reasonable consistency turn this down, even if it was way out there in Menlo Park?

Lick moved very swiftly. By early 1963 we had a funded project. But, whereas I had proposed using a local computer and building an interactive workstation, Lick asked us instead to connect a display to

the System Development Corporation's (SDC's) AN/FSQ32 computer, on site in Santa Monica, to do our experimenting under the Q32's projected new time-sharing system. (Converting the Q32 to be a time-shared machine was SDC's IPTO project.)

Later that year, our project was modified to include an online data link from Menlo Park to Santa Monica, with a CDC 160A minicomputer at our end for a communication manager, supporting our small-display workstation. For various reasons, not uncommon in pioneering ventures, that first year was very unproductive relative to the purposes and plan of our project. Lick was willing to put some more support into the direct goal (more or less as originally proposed), but the support level he could offer wasn't enough to pay for both a small research staff and some interactive computer support. Mind you, the CDC 160A, which was the only commercially suitable minicomputer that we knew of, even though having only 8K of 12-bit words, and running at about 6 microseconds per instruction, cost well over \$100,000 (1963 dollars). It had paper tape in and out; if the system crashed, you had to load the application program from paper tape, and the most recent dump of your working file (paper tape), before you could continue. A crude, industry-standard Flexowriter (online typewriter) could be driven; otherwise it was paper tape in and out.

What saved my program from extinction then was the arrival of an out-of-the-blue support offer from Bob Taylor, who at that time was a psychologist working at NASA headquarters. I had visited him months before, leaving copies of the Framework report and our proposal. I had been unaware that meanwhile he had been seeking funds and a contracting channel to provide some support. The combined ARPA and NASA support enabled us to equip ourselves and begin developing Version 1 of what evolved into the NLS and AUGMENT systems. Paul Fuhrmeister, and later Eugene Gribble of NASA's Langley Research Center, had to stick out their necks as successive heads of Langley's large computational division to support the direction and supervise NASA's support for our program, which continued several years after Taylor left NASA to join ARPA's IPTO office.

Our ARPA support grew and was fostered by Lick's successors—Ivan Sutherland, Bob Taylor, and Larry Roberts. Meanwhile, the Air Force's Rome Air Development Center, at Rome, New York, began to supply supporting funds. By 1967, it was recognized that the respective contributions from ARPA, NASA, and RADC represented significant parts of a coordinated program. The other agencies began funneling their funds through RADC, which served for many years to both monitor and manage our contracts, as well as provide their own significant share of support funds. John McNamara and Duane Stone provided strong support and contract liaison from RADC.

NASA support ended by 1969, and ARPA and RADC provided

significant support until 1977, although from 1974 the funding became even more for supporting applications and developments for other organizations, for targets formulated by others (e.g., the National Software Works). The continuing pursuit of augmentation along my strategic vector virtually stopped.

The Augmentation Research Center (ARC)

An historically important organizational cluster emerged at Stanford Research Institute in the 1960s, peaked about 1974, and was scattered in 1977—with a small core carrying forth in a commercial (and then industrial) environment to the present. It grew by ones and twos from 1963, as it collected “permanent” members from the SRI technical staff, and recruited new ones from the outside. By 1969 I believe we were about 18 strong; this grew steadily until by 1976 we totalled about 45. In 1973 we made two explicit subgroups, one headed by Dick Watson doing development of software (and some hardware), and one headed by Jim Norton handling operations and applications support.

SRI was organized by divisions, each containing a group of laboratories, the hierarchy being formed according to the associated disciplines. ARC grew to laboratory size and status, but it became something of a problem for SRI. Other laboratories (at least in science and engineering) operated more or less as a “farmers’ market,” where small and changing clusters of researchers promoted and conducted research projects as a loose federation. The management structure, budgeting, accounting, and financing for the Institute had evolved to support this kind of business. But ARC was driven by a coherent, long-term pursuit. This involved the continuing evolution of an ever-larger and more sophisticated system of hardware and software. It also came to involve delivering solid support service to outside clients to provide meaningful environments for learning about the all-important process of coevolution between the human-system and tool-system components of our organizations (as per my conceptual framework).

It didn’t seem unreasonable to me to pursue this course; things similar and on a grander scale are common for other researchers. It is taken for granted, for example, that funding agencies will build and operate accelerators and observatories in support of research in nuclear physics and astronomy, or will outfit ships and airplanes to support research expeditions. But whatever my perception, there were some significant problems and stresses with which our over-all environment didn’t have effective ways to cope. In the particular dynamics involved, there were probably seven relevant parties: me, the ARC staff, other SRI researchers, SRI management and administration, ARC’s sponsors, ARC’s utility-service clients, and other groups of researchers outside of SRI. It would be an interesting historical study to try to un-

derstand the diversity of perception that must have existed among this set of players. What did the different parties perceive for the future of workstations, for the range of function and application that would come about, for the systems architectures and standards that must emerge, and for the impact on the organizations that learned how to harness these most successfully?

Even as a central party in what happened, I've not understood the dynamics. But I am pretty sure that disparities among the perceptions of all of the above parties had a major part in what to me was the "great collapse of SRI-ARC." Even if I had done everything right over the years (a laughable hypothesis), it is now fairly clear to me that it isn't the market's fault if someone fails in trying to sell it something that the market isn't ready for. In other words, I can't blame those other groups. (Which of course makes for a personal problem, since during those times of black discouragement when one wants desperately to blame someone, there is only one candidate—that guy at the head of the list.)

In 1977, SRI judged it better to move our large-system development and external-service activities out from the research institute environment and into a suitable commercial environment. They advertised, entertained prospective bidders, made a selection, and negotiated a transfer of the business to Tymshare, Inc., of Cupertino, California. The system was renamed AUGMENT and marketed as part of Tymshare's integrated Office Automation services. In 1984, McDonnell Douglas Corporation acquired Tymshare, and the small AUGMENT business is now operated as the Augmentation Systems Division of the Computer Systems Company within the MDC Information Systems Group.

SOME OF ARC'S EARLY PRODUCTS—1964 TO 1968

Let's take a look at some of the actual experiments we tried during the years from 1964 to 1968 at SRI. Around 1964 we got off the CDC-160A system, which in fact we were using as a personal system. This was our first, real, stand-alone machine.

Screen Selection Tests and the Mouse

In those days the cost of getting display systems to work was very high. My strategic plan under the above-mentioned framework was to skip the then-prevalent, interactive typewriters and focus from the outset on displays. My assumption was that there is a great deal to learn about how to harness highly interactive display capability, and by the time we really learned how, the prices would be down considerably.

We wanted to start early experimenting with screen selection. The idea of working and interacting very actively with the display meant

that we had to tell the computer what we were looking at, so we needed a screen selection device. There was a lot of argument about light pens and tracking balls in those days, but none of those arguments served our needs very directly. I wanted to find the best thing that would serve us in the context in which we wanted to work—text and structured items and interactive commands.

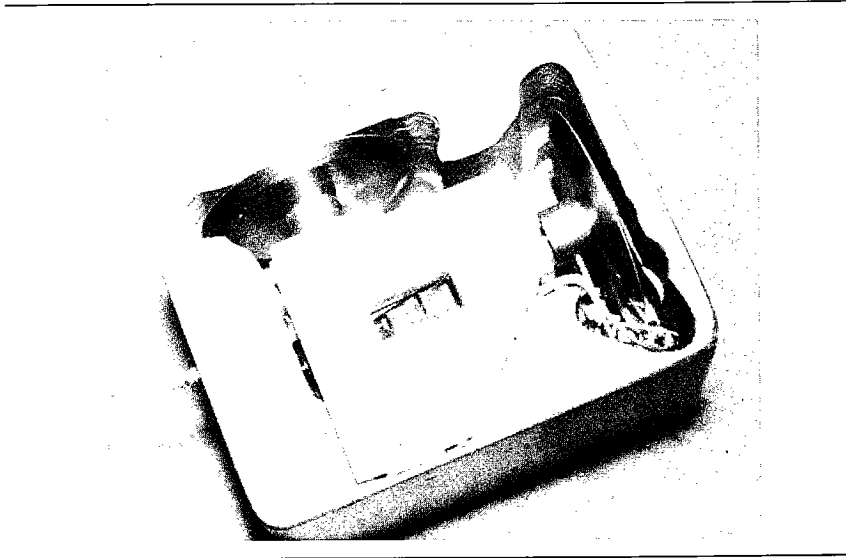
The context was important. So we set up computer-controlled experiments oriented to our working mode, where we assumed that purposeful knowledge workers would be spending a significant portion of their time writing, studying, analyzing, and even debugging. We collected a set of candidate screen-selection devices to test. We did our experiments with our one workstation of that period, which is shown in Fig. 1, together with one of our test devices. In trying to be complete about the array of test devices, I dug up some old notes of mine describing a possibility that eventually turned into the very first mouse (Fig. 2). The tests were carefully run, and we even integrated selection errors and their correction penalties in evaluating the "goodness" of a device. The experiments and their results were fully reported,⁵ and later described in a paper.⁶ The graph in Fig. 3 is representative of our results.

The mouse consistently beat out the other devices for fast, accurate screen selection in our working context. For some months we left the other devices attached to the workstation so that a user could use the device of his choice, but when it became clear that everyone chose to use the mouse, we abandoned the other devices.

FIGURE 1
CDC3100 workstation
at which we
experimented with
selection devices.



FIGURE 2
Bottom of wooden
"mouse," one of the
first selection
devices.



No one is quite sure why it got named a "mouse," or who first started using that name. None of us would have thought that the name would have stayed with it out into the world, but the thing that none of us would have believed either was how long it would take for it to find its way out there.

We also experimented with a way to move your knee up and down or swing it sideways to move the cursor, so you could keep your hands on the keyboard (Fig. 4). I also built something that allowed one to control the cursor movement by rotating your head for sideways cursor control, or nodding your head up or down for vertical cursor control. This "nose-pointing control" of the cursor left both hands free

FIGURE 3
Cursor-select test
graph comparing
various selection
devices.

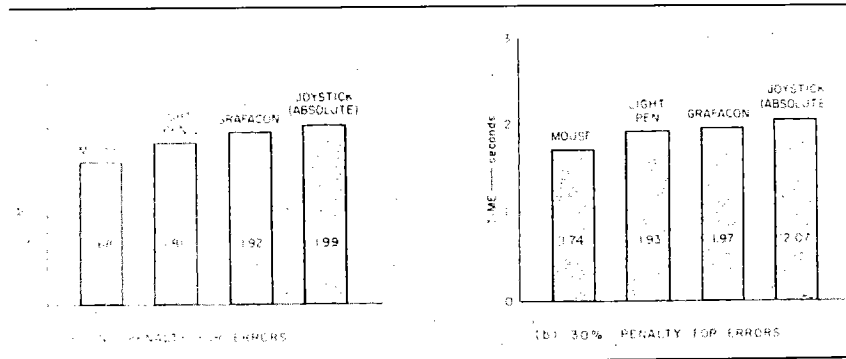
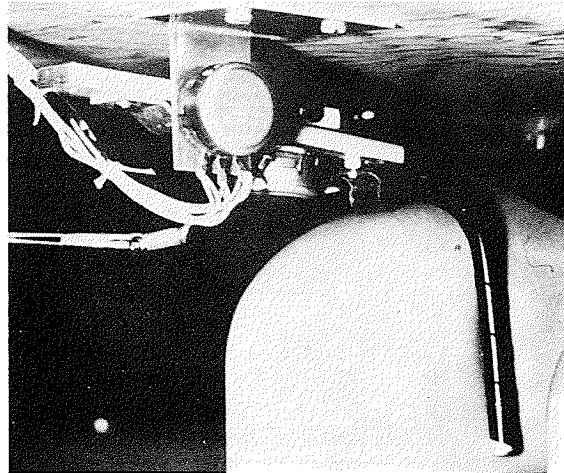


FIGURE 4
An experimental
knee control device.

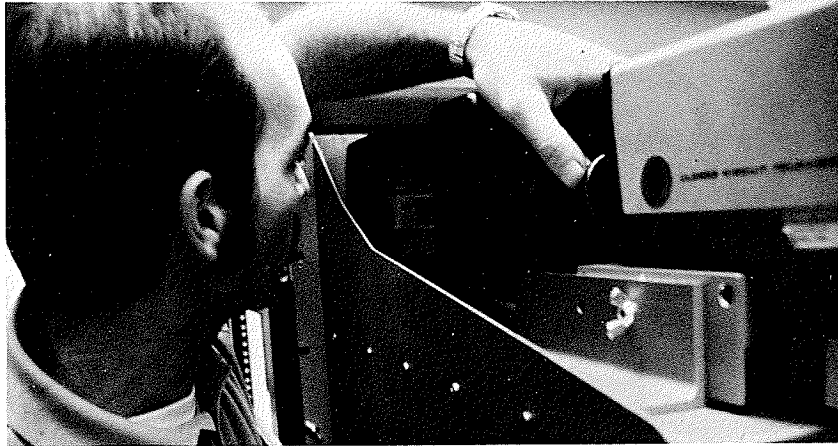


to operate the keyboard. In each case, some muscles would cramp up. Really, in almost any of these cases, you can get used to things like that. I'm sure there are a lot of things to explore in the future about what is going to work best for interfacing, so that you can tell the computer to which objects on the screen you want to direct its attention. However, we had many things to do, so it was a diminishing return for us to pursue these alternative devices—we stayed with the mouse. I have always assumed, though, that something better than the mouse is likely to emerge once the user market becomes more adventurous in exploring the means to higher performance.

OUR MULTI-USER COMPUTER AND DISPLAY SYSTEM, CIRCA 1968

The first time-sharing system we got was an SDS940. We considered a number of alternative ways to provide ourselves with a flexible, responsive display system, and finally designed our own. The computer and display drivers time-shared their attention among multiple, 5-inch CRTs, which happened to be the most economical size for a given percentage of screen resolution. In front of each CRT we added a commercial, high-quality video camera, mounted with a light shroud over the camera lens and CRT screen (Fig. 5). The resulting video signal, amplified and piped out to our laboratory, drove the video monitors that were our workstation displays. Two display generators, each driving up to eight CRTs, implemented with vacuum-tube technology were both bulky (Fig. 6) and very costly. It took one and a half people to keep those things running all of the time. The stroke-generated characters and vector graphics allowed us to have flexible, mixed text and

FIGURE 5
Our early computer
set up with 5-inch
CRTs.



graphic document presentations. The display generators were driven from a direct-memory-access channel that provided very fast (i.e., one refresh-cycle time) creation of a new display image. Figure 7 is a picture of one of our workstations—the TV monitor, with a little wooden table we made and a keyboard and a mouse and a keyset.

We explicitly designed for a detached keyboard. My workstation-design philosophy was to fix it so what you are looking at is positioned for best viewing, and the devices you use to control the computer should be located where it is best for you to operate them. Don't get caught in the anachronism that because we got used to paper and pencil and that technology, we ought to be able to have our controls right

FIGURE 6
The Augmentation
Project display
system.

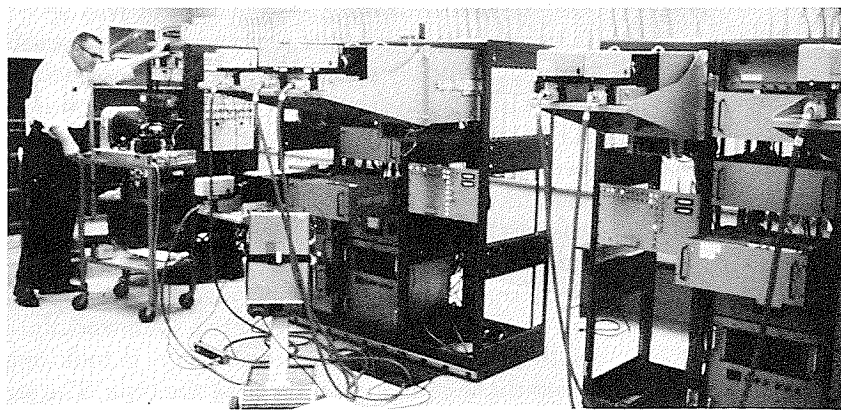
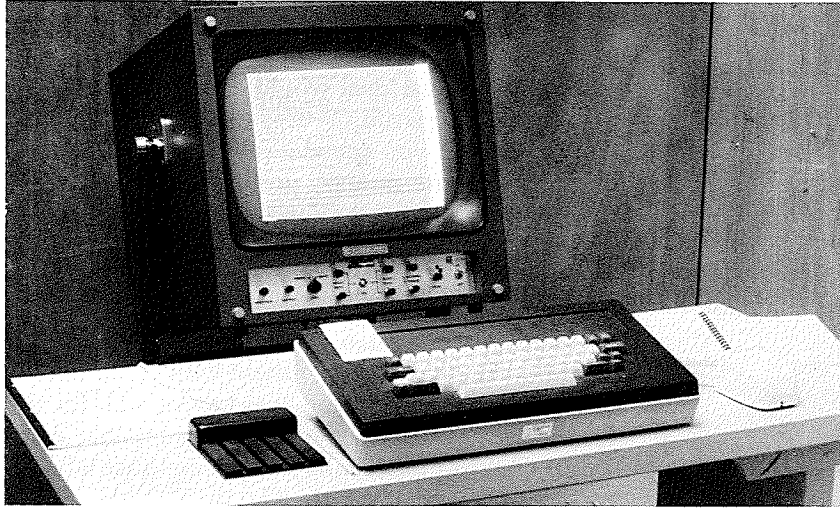


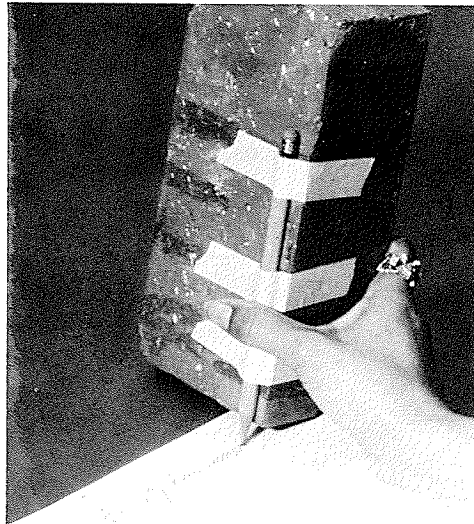
FIGURE 7
Basic workstation:
table with keyset,
detached keyboard,
mouse, and a
separate monitor.



on the surface of the thing that we're working on. It may end up that that way is best, but don't make an a priori assumption. So we didn't.

One way of explaining to somebody why it could make a significant difference if you can do things faster, is to provide a counter example. So, I had them write with a brick taped to their pencil (Fig. 8), because it's only a matter of happenstance that the scale of our body

FIGURE 8
Brick "writing"
device.



and our tools and such lets us write as fast as we can. What if it were slow and tedious to write? A person doesn't have to work that way very long before starting to realize that our academic work, our books—a great deal would change in our world if that's how hard it had been to write.

What if you speed it up? The keyset shown in Fig. 9, in combination with the mouse, provides a two-handed, higher-speed option. When you strike a chord, the computer interprets it as a character—not really distinguishing between characters generated by keyset or keyboard. Our chord-character code (a binary counting scheme mapped to the alphabet) is not really very difficult, my six-year old kids took less than a week to learn it. At SRI, we had a project in the early 1960s to experiment with computer-aided, psycho-motor skill training, and we used this keyset skill for the very first thing to be learned. The experiment blew up because everybody learned it so fast we couldn't differentiate between those who did and those who didn't have special computer aids.

Figure 10 is sort of the picture that I think about when I'm told that we've got to make it easy to learn. Tricycles are "easy to learn and natural to use"—no hard balancing problems. How much is it worth to you to extend your mobility? A few hours of learning to balance on two wheels?

By 1968 I had a workstation like the one in Fig. 11 in my office, and I just started "living" on it. In our lab we had six to ten more, time-shared among our group. The whole feeling was great. It was a very, very interesting sort of activity in those days. Who says we didn't do workstation research? This one, our "Yoga Workstation" (Fig. 12 with Bill Duval), got to be one of the favorites for some reason. This is what you saw all the time—people sitting there in their two-handed working mode, occasionally switching to type on a keyboard.

Figure 13 shows Bill English with a workstation made for us by the Herman Miller office furniture company, who were trying to experiment with us. This little console swivelled on one of the chairs and

FIGURE 9
Keyset input device.

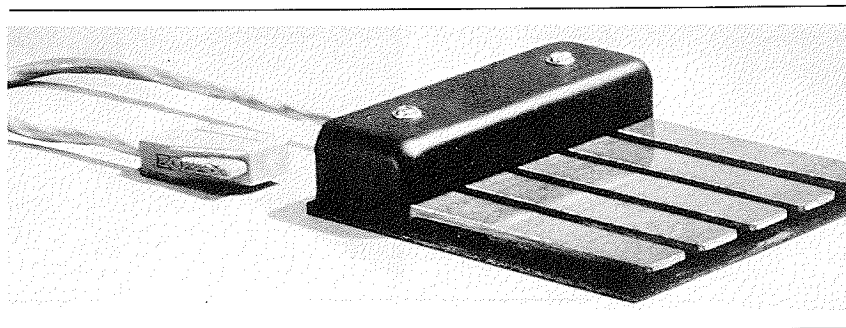
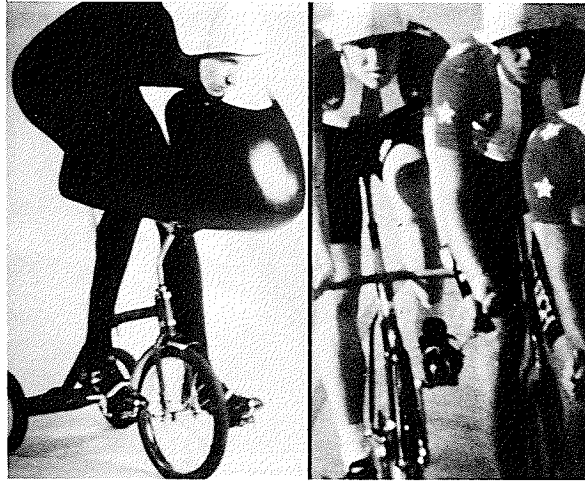


FIGURE 10
Trike/Bike.



went where you went; you could lean back and work with it very nicely. I wanted to show this workstation and Bill English, who deserves an immense amount of credit for getting things to work. I waved my hands and pointed, and somehow didn't make much progress. But Bill really made things work in those days, so I owe him a lot.

FIGURE 11
Doug's office
workstation
replicated 6-10 times
in the lab.

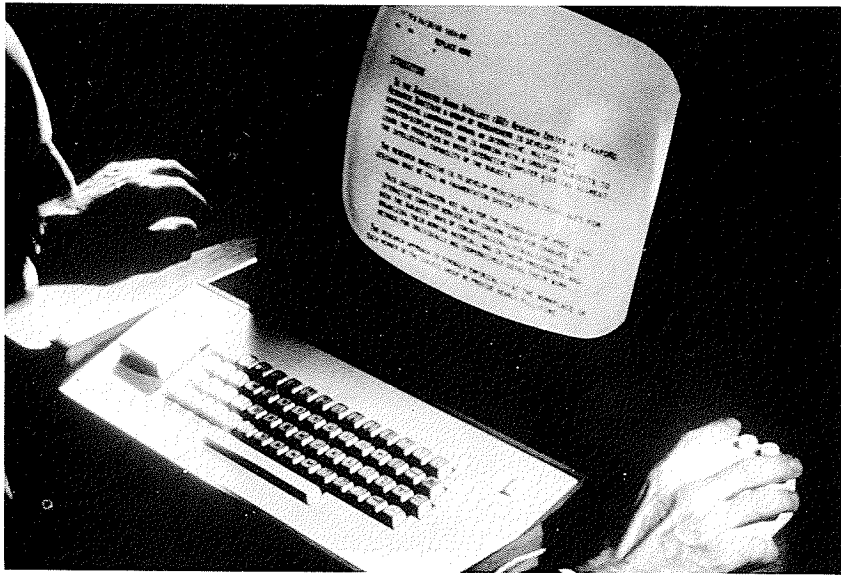
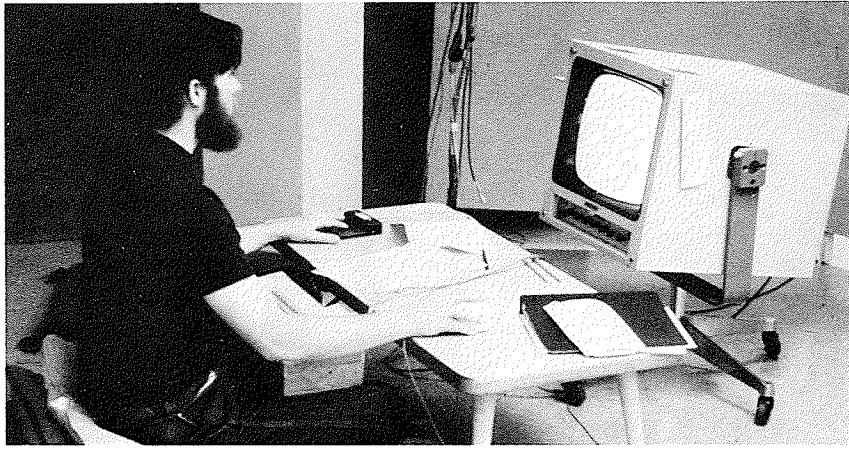


FIGURE 12

Yoga workstation—
Bill Duval is pictured
here.



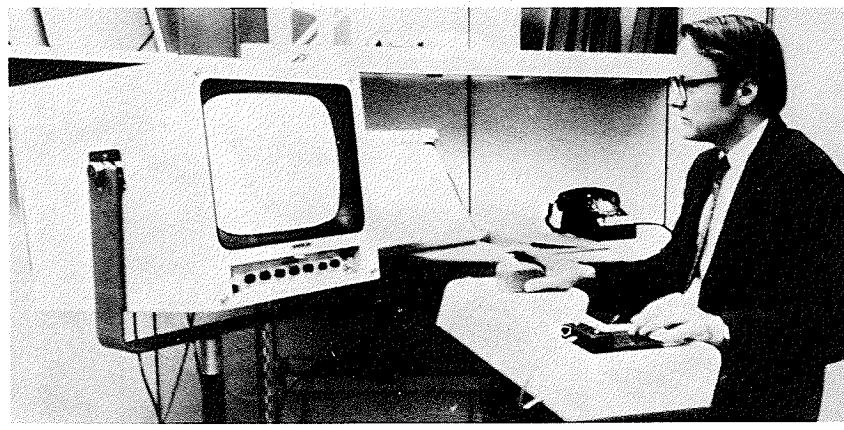
OUR "BIG SHOW"—THE 1968 FALL JOINT COMPUTER CONFERENCE

By 1968 we had a marvelous system. We called it "NLS" (later "AUGMENT"). "NLS" stood for "online system," in contrast to "offline system," which we dubbed "FLS" just to have different acronyms. A few people would come and visit us, but we didn't seem to be getting the type of general interest that I expected. I was looking for a better way to show people, so we took an immense risk and applied for a special session at the ACM/IEEE-CS Fall Joint Computer Conference in San Francisco in December 1968.

We set up to give an online presentation using a video projector pointing at a 20-foot screen. Brooks Hall is a large auditorium, and that

FIGURE 13

Bill English at our
Herman Miller
designed
workstation.

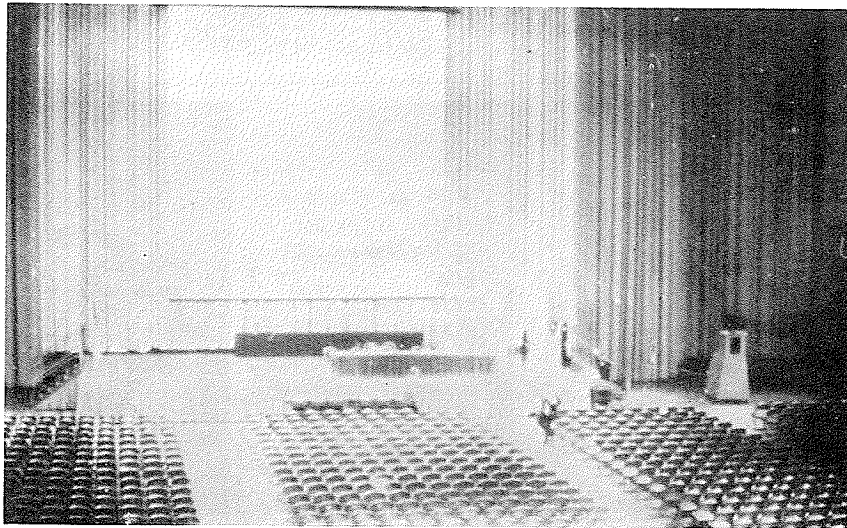


video projector could put up our display images so you could read them easily from up in the balcony (Fig. 14). The video projector we rented (built by a Swiss company, Eidophor) used a high-intensity projection lamp whose light was modulated by a thin film of oil, which in turn was modulated by the video signal. On the right side of the stage, I sat at our Herman Miller console. We set up a folding screen as a back drop behind me. I saw the same image on my workstation screen there as was projected for the audience to see.

We built special electronics that picked up the control inputs from my mouse, keyset, and keyboard and piped them down to SRI over a telephone hookup. We leased two microwave lines up from our laboratory in SRI, roughly 30 miles. It took two additional antennas on the roof at SRI, four more on a truck up on Skyline Boulevard, and two on the roof of the conference center. It cost money—running that video projector, and getting people to help us do all of that, cost money; making the special I/O cost money; and leveraging special remote-presentation technology on top of our advanced, developmental laboratory technology created extra risk—and I was using research money. The nice people at ARPA and NASA, who were funding us, effectively had to say, “Don’t tell me!” because if this had flopped, we would have gotten in trouble. But anyway, it worked, and the main reason was because of Bill English’s genius for making things work.

Back in our lab, we dismantled a number of the display units in our display system, so that we could use the cameras in San Francisco and SRI. We borrowed a few tripods and got some extra people to be

FIGURE 14
Auditorium of the
ACM/IEEE-CS Fall
Joint Computer
Conference,
December 1968.



camera people. One of our friends, Stuart Brand, who was at that time working on his first *Whole Earth Catalog*, helped as well. So it was really a group project; there were about 17 of us.

On my console on the stage, there was a camera mounted that caught my face. Another camera, mounted overhead, looked down on the workstation controls. In the back of the room, Bill English controlled use of these two video signals as well as the two video signals coming up from SRI that could bring either camera or computer video. Bill could select any of these four video images with optional mixing and frame splitting. We had an intercom that allowed him to direct the action of the people in our lab at SRI who were generating computer images or handling the cameras sending the video up from SRI.

We didn't use any specially made system capabilities; we were just using NLS the way it worked at that time. It had mixed text and graphics, so we could use those to display and represent things. We had the agenda in NLS, and we could run different parts and show diagrams; we could do things as examples. So it was a mix of things of: here's the script and stuff to tell you about, and here's the way it runs; we could also bring other display screens or faces, from our SRI lab, in and out on the screen. At that time we firmly imagined that this was the way future conferences would be run.

We could do screen splitting. Figure 15 shows the agenda list with a little marker to show that I'm between two particular items. In Fig. 16, I make a temporary (shopping) list. This was the beginning of our demonstrating ways of structuring ideas. The NLS system supported the user in getting the list organized into categories.

We wanted to show how the mouse works. The projected video showed Don Andrews controlling a cursor from our SRI laboratory by

FIGURE 15
Still frame from the
FJCC '68. Split screen
with Doug on the
right and demo
agenda on the left.

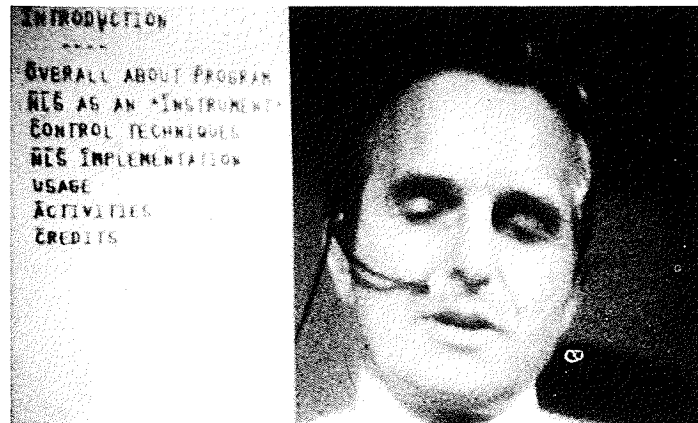
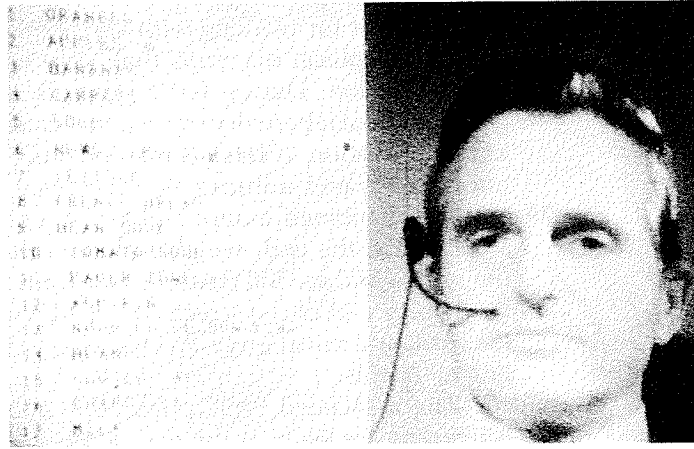


FIGURE 16

The text is now an unstructured shopping list.



moving his mouse around. The superimposed video image of the display screen showed that the cursor would follow it exactly to show how the wheels worked. Remember, this was 1968—the first public appearance of a mouse. I could also show how the simultaneous use of mouse and keyset worked in such a way that the audience could watch my hands in the lower window and see the computer action in the upper window.

Then we brought in Jeff Rulifson to tell about how the software works (Fig. 17). At the same time, his face could be brought in and out

FIGURE 17

Jeff Rulifson can be seen on the screen mixed in behind code of a procedure.



behind the display image that he was working with, demonstrating NLS's power for working with very explicitly structured software. He showed graphical diagrams that were embedded in the source-code documentation. During Jeff's presentation, Bill English brought the picture from a laboratory camera that caught the view (Fig. 18) of Jeff's keyset operation as he was manipulating his demonstration images—unconscious and unhurried—a nice way to show the fluid speed offered by combined mouse and keyset use.

Toward the end, we also showed that we could cut a "hole" in the screen and see Bill Paxton's face from SRI (Fig. 19). For the computer-display part of the screen, we could switch back and forth between his work and mine; and we could also switch which of us was controlling all of this.

The associated FJCC publication about NLS⁷ and other relevant references are listed at the end of this paper.

[*Editor's Note:* Engelbart's colleagues—Bill English, Charles Irby, Jeff Rulifson, Bill Duval, and Bill Paxton—all found themselves at Xerox PARC in the 1970s, creating a personal workstation that embodied these ideas.]

NLS ENHANCEMENTS—INTO THE MID-1970s

You can't imagine the relief when it worked. It went on for 90 minutes, and afterward we thought for sure that the world would be talking about everybody starting to augment now. Well, it didn't happen, but we went ahead anyway. I want to discuss a few of the things we did

FIGURE 18
Jeff's keyset and hand behind some lines of text.

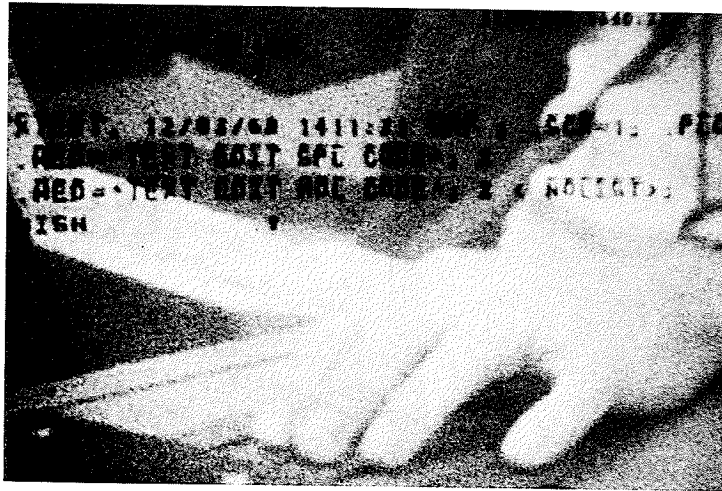
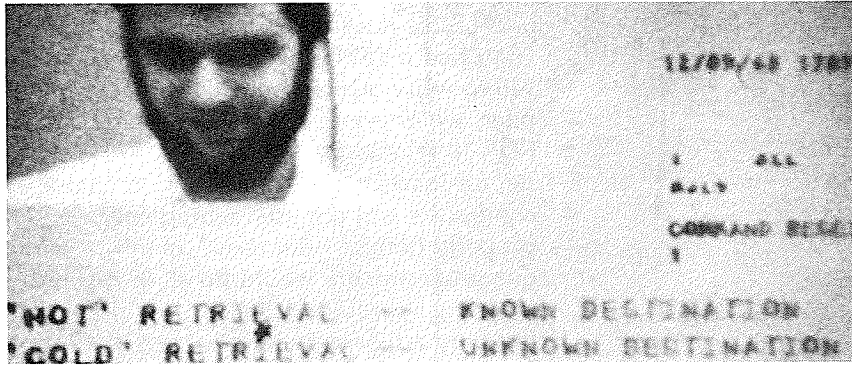


FIGURE 19
View of text retrieval
with image of Bill
Paxton in
background.



physically to the system after that, and then go into some of the conceptual framework.

1969

- We began design of windowing capability for NLS.
- We developed concept of a user "reaching through" his personal work place (i.e., his familiar online working files and application programs) to access less basic, specialized data and application processes (and other people); that is, the "reach through" should provide access to these, translated by the integrated support system, so as to appear as coherent parts of his familiar, personal work place.
- We specified our first mail and "Journal" system as part of an explicit pursuit of a "Dialog Support System," planning for it to be part of our ARPANET-NIC service.
- We developed document-outputting capability processing our composite, text-graphic document files to drive a service-bureau, CRT-based, full-page, Stromberg-Carlson photo printer to produce documentation with graphics and text mixed on the same pages.
- We became the second host on the ARPANET with our SDS 940. (UCLA was first, UCSB next, then the University of Utah, then. . . .)

1970⁸

- Detailed use of NLS began for internal management processes of ARC: cost records, working forecasts, purchase requisitions, and so on.

- We began using the ARPANET to facilitate our reprogramming of NLS for the forthcoming PDP-10 TENEX. The University of Utah had a TENEX on the network, and we used NLS on the 940 to write our new PDP-10 code; using our tree-meta compiler, we developed a cross-compiler for our 940 that produced PDP-10 relocatable binary code. We would ship that over the net for loading and debugging on Utah's TENEX. When the two computers and the intervening network link were all working properly (lots of "flat tires" as in the early days of automobiles), our programmers would do all of this back and forth transitioning "through" the same workstation. I think that it was not only a record-making way of working, but the NLS transport task was accomplished in remarkably short time (we attributed part of the efficiency to the network, and part to the use of NLS).
- We brought NLS up on the PDP-10 TENEX with improved and new features (including multiple windows). The transfer process, and a detailed description of the design changes and new features for NLS are described in a June 1971 technical report.⁹
- We began using our Mail/Journal system within our group. Integrated into NLS, this assumed that a mail item was a document—so any part of all of an NLS document could be sent—and provided for a permanent record in explicitly retrievable form (our Journal). As an electronic-mail system, this was quite advanced. It had a directory service (our Ident System) to provide mail-relevant information about registered users; mail distribution was addressed by people's Idents, with no need to know or specify which host they used. Fields were provided for superseding other items, and for attaching keywords. An on-line index was provided for stored items.¹⁰

1972

- We began developing our first, integrated Help system.
- We formulated the "AKW Architecture," implemented in stages.¹¹
- We implemented the "shared-screen," televiewing mode of on-line collaboration between two or more NLS users.¹²

1973

- We brought up a table subsystem in NLS.
- We designed our first, totally modular user interface system, as later described in the *OAC '82 Digest*,¹³ and got it running on

a PDP-11 that talked to our TENEX through the network, via our procedure call protocol.

- We developed our line processor, as described by Don Andrews.¹⁴ It incorporated Intel's first microprocessor (the 4004) in a special box that was inserted in the communication line between a dumb display terminal and a modem. This made use of our virtual terminal protocols, and managed a multi-window, two-dimensional screen using off-the-shelf, "dumb" display terminals. Our mouse and keyset input devices were plugged into the line processor, which appropriately translated their actions to control cursor position and special communications to the host. A printer port on the line-processor provided local printout service; a special communication protocol allowed the host to send printer packets mixed in with display-support packets.
- We finalized specification for our network virtual terminal, something that has become a key part of our architecture. The objective, on the one hand, was to free the application programmers from worrying about the special features of different workstations, and on the other hand, to enable more flexible evolution by users of workstations they may adopt to fit particular needs. As part of this, there was a terminal-independent display manipulation protocol for communication from application program to terminal, and an application independent input protocol for communicating from terminal to application program.
- We generalized the file structure of our document files to provide for generalized property structures associated with each addressable object, intended to accommodate composite integration of such as graphics, digitized speech, scan-coded images, or any other arbitrary data form.

1974

- We gave up our high-performance, local display system for the line-processor supported, remote display system—to make ourselves live with the same remote services as our NIC clients and Utility customers. (On principle, we gave up our integrated, direct-view graphics and the fast response of our direct-memory-access, local display generator.)
- We opened our "Workshop Utility Service." Delivering NLS service over the ARPANET to DOD customers as pilot applications of office information service. We had gone out on bid for

commercial time-sharing services, selected Tymshare Inc. of Cupertino, Calif.; their host, named Office-1, provided the computer service. We fielded special trainers and application development staffs and cultivated special customer representatives into a spirited community.

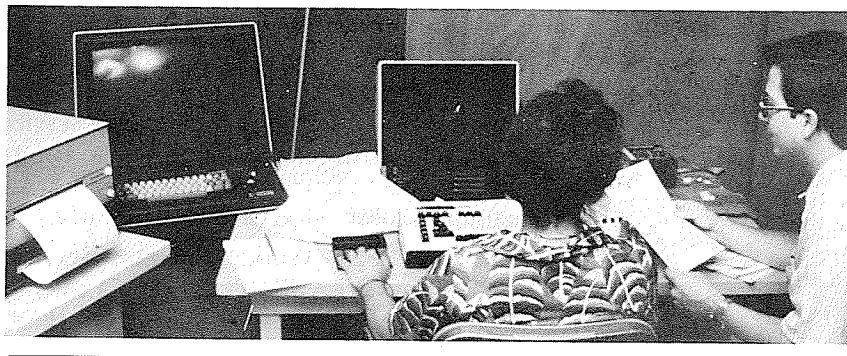
1975

- We implemented our new, integrated graphics system, which could support remote display and manipulation of illustrative graphics on a Tektronix 4014 storage-tube display plugged into the line-processor's printer port. Figure 20 shows the graphic-station setup used for development in our lab. Bob Belleville, to the right in the picture, developed this "new" graphics software. (He subsequently went to Xerox and helped them with the Star hardware, and then was the project manager for the Apple MacIntosh.)

In 1973 we had generalized our file structure with provision for attribute-value property structures associated with each hierarchic node. We could then embed arbitrary data objects in our documents. The first utilization was to reinstate embedded graphics, extending our hierarchical list structure and its pointers to the text objects with, for example, an associated pointer on one of the text objects to a graphic substructure for the associated illustration. The user's concept would be shown in an associated "document-page" image. We assumed that soon we would be using digitized speech strings and executable object code as part of our composite documents. And some day, when storage would be cheaper, we would even be embedding scanned images.

An illustration produced as a plotter-driver file by any other graphics system could be picked up and attached to a specified location in an NLS document, and could be subsequently viewed and modified

FIGURE 20
Bob Belleville at Tek
graphic workstation.



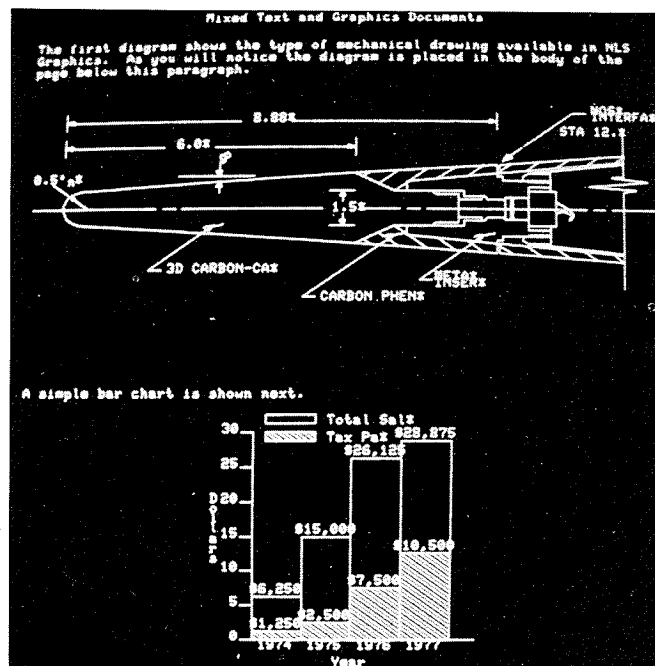
with the NLS Graphics Subsystem. Figure 21 shows two such "ingested" illustrations, as viewed in two adjacent NLS graphic windows. Our use was oriented toward "illustrative graphics," and we had a reasonably complete set of capabilities for construction and manipulation.¹⁵

By this time, we had a well-developed Output Processor (our "document compiler"), which acted upon a large vocabulary of embedded-text directives to provide font selection, columnation, running headers and footers, and much more.¹⁶ In conjunction with the Graphics Subsystem, the Output Processor was enhanced so that a user could direct that a properly scaled image of any graphic illustration be located appropriately within a multi-font page layout.

All this was available by 1975-1976. But the problem at the time was that for somebody to use this, they would have to buy a \$10,000 or \$15,000 storage tube terminal to go with the line processor and text terminal. We couldn't get past this business of "if they don't have it, they don't know they need it; and if they don't know they need it, they don't want to buy it." It was a little difficult at the time, so the graphics sort of atrophied until recently.

The whole approach was that our files were document oriented—documents in a very general sense. These are what contain the descrip-

FIGURE 21
Tek view of mixed text-graphic, two-column page, produced with NLS.



tions, arguments, proposals, whatever, the things you're trying to map your thoughts and arguments into to form a communication.

Another thing we did, though, in 1967, was to help people collaborate in face-to-face meetings. Here again, Bill English whipped it all together very fast—special small narrow tables that you could work around and sit at in a conference meeting, where the monitors were down low enough so you could see over them and see each other well. Each monitor had the same image that was brought from our one-user 3100 at the time, but we had one workstation and some mice. Anybody who wanted to pick up a mouse and push the button could, and this would activate a large, special cursor that could be controlled and moved around, so any participant could point out things on the common display. We had a review meeting among our sponsors at the time. The picture (Fig. 22) shows Bill English, Don Andrews, Dave Hopper, and Barry Wesler, who was Bob Taylor's assistant. Bob was in attendance at the time, too, participating in this conference.

Another thing we did by 1970 was to bring up our electronic mail system as part of our collaboration. There was really a very sophisticated mail system with user identification, catalogs, and all sorts of fields that you could use to send and answer things. One extra feature was that you could send whole documents; it didn't matter whether it was a little one sentence note or a whole document.

A document could go in and be catalogued into a system called the "Journal," which was similar to the idea of pasting it down on a table (Fig. 23). We had linkages and internal addressability, so an embedded link in one document could directly cite an arbitrary pas-

FIGURE 22
1967 multi-CRT
conference room.
Seated are Don
Andrews, Dave
Hopper, Bill English,
Doug, and Barry
Wexler.

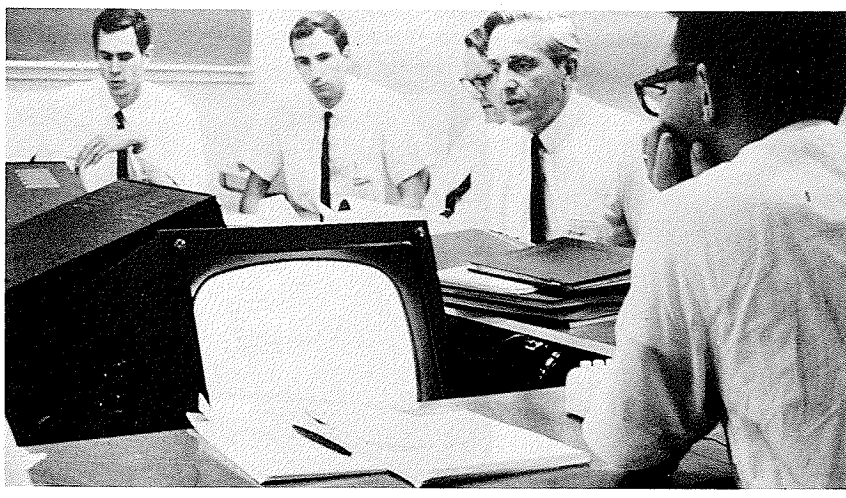
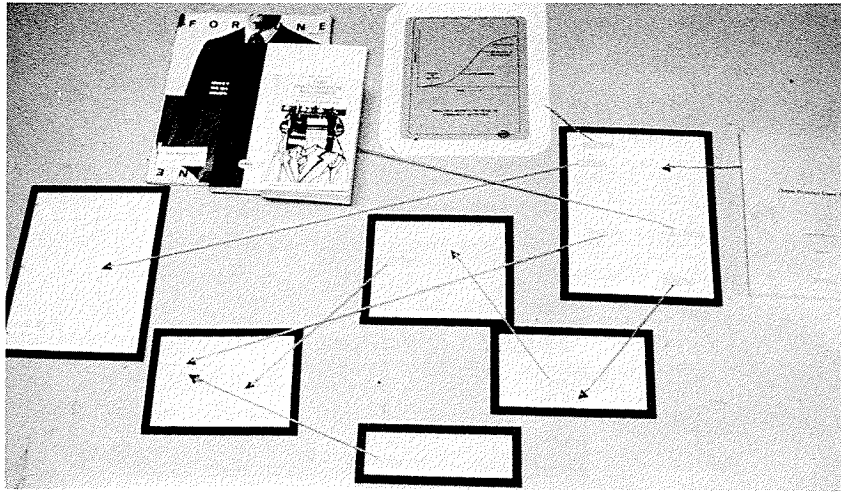


FIGURE 23
Six-document table
model of a "Journal"
with two linked
document items.

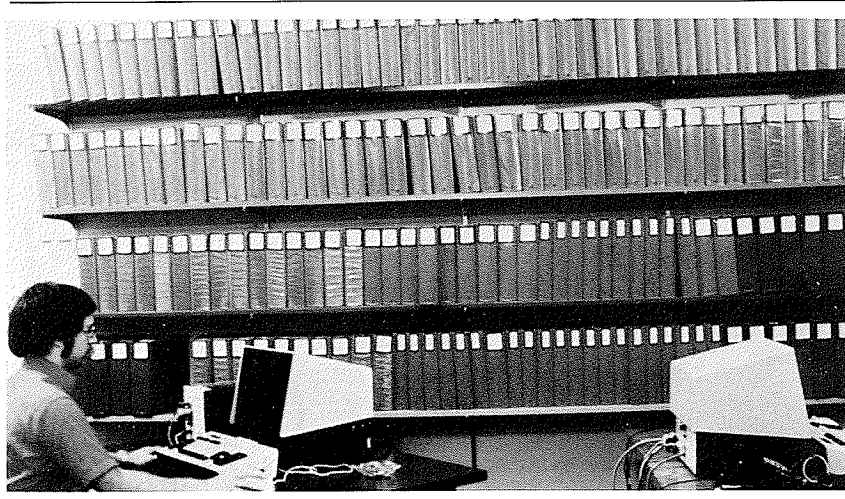


sage in any other. Successive documents could be entered in the system and easily be cross-referenced back to each other. This supported what we called our "recorded dialog"—an important part that we assumed was needed for a community of people to work together effectively. It is an extremely powerful capability.

We originally designed our Mail/Journal system to give the user a choice as to whether to make an entry unrecorded (as in current mail systems), or to be recorded in the Journal. I knew that there would be lots of question, and some quandry, associated with the question "to record or not to record." I assumed that many fewer items would be recorded than "should be"—as might be judged after we someday would learn the value and establish criteria for recording. So, I ordained that *all* entries would be recorded—no option. It put some people under considerable strain. I think that one person actually never could bring himself to enter a memo into the Journal—the idea that it was "forever" stopped him cold. Another person, a very valuable contributor, somehow felt violently opposed to the basic concept, and it possibly hastened his departure. But after a year or so, it was used as a matter of course by essentially everyone. There were interesting and unexpected payoffs.

Up until about 1975, we made a practice of printing out every one of the stored documents—mostly as an alternate analytic process to watch the dynamics of Journal use. The documents are stored by number, in binders as shown in Fig. 24. This indicates how big our Journal collection got in about four or five years. The number now for that particular journal is well over 100,000 items. We have an arbitrary

FIGURE 24
Hardcopy of the
Journal filled four
shelves.



number of other journal systems that customer groups can install and administer themselves, a very powerful potential archive for collaborative work.

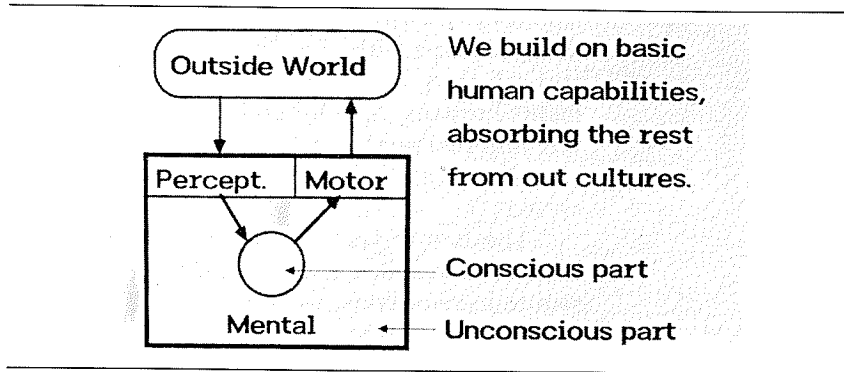
FRAMEWORK—TO 1968 NLS

Let's now shift back to the conceptual framework, originally documented in a 1962 paper.¹⁷ I had this immensely intuitive feeling that humans were going to be able to derive a great deal of capability from computer systems. I had very real images in my mind of sitting at a display console, interacting with a computer, seeing all sorts of strange symbology coming up that we could invent and develop to facilitate our thinking. We would no longer be limited to working with paper and other such laborious means. Other people could be sitting at similar consoles tied to their machine and we could be collaborating in brand new, exciting ways. We could be doing all sorts of things to control a computer.

At that time, even though I didn't know how the innards of a computer worked, I had enough engineering background to know that if a computer could read a card, it could sense keys and any other action I might want to do. If it could drive a printer or card punch, it could put whatever I wanted onto a display. What I didn't understand at the time was the economics and all that, but I said, "Look, I've got a whole career ahead of me, let's go after all of this."

By 1959 or so, I got a chance to sit down and say, "How would that really work? What are the basics?" As an engineer, I ended up with a simplifying model (Fig. 25). Here's a human wanting to do this

FIGURE 25
Starting to think about augmenting the human intellect by beginning with basics.

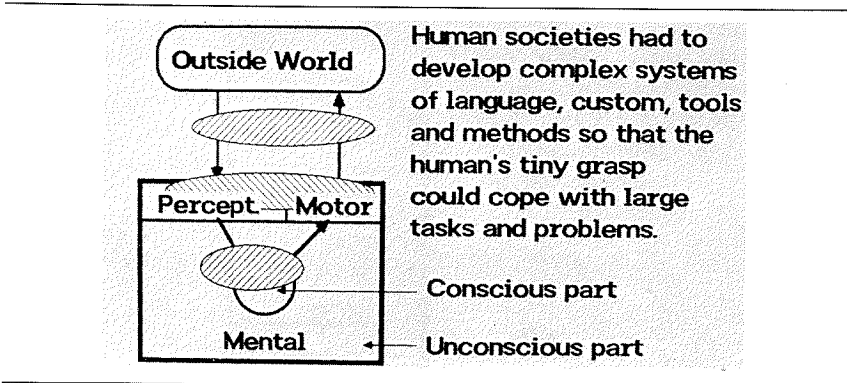


knowledge work; he's got capabilities within his skin that we can make use of, a lot of mental capabilities we know of, and some of it he's even conscious of. Those are marvelous machines there—motor machinery to actuate things in the outside world, and sensor and perceptual machinery to get the idea of what's going on. And that's what he has to support his interaction with the outside world.

How can we improve the human's capability (Fig. 26)? Well, it turns out that there have been a lot of people working on this problem for many generations. And we've got a whole "culture-full" of things that we are indoctrinated and trained into, both conscious things and unconscious things. We have lots of skills, motorwise and perception wise, that we're not even aware of. For example, just how long do you think it took to learn to brush your teeth? It takes quite a while; it's a skill. These sensory-motor things all developed in order to help us interact with the layers of other things that our culture provides.

Many such things are available to us, tools and methods that let us live within a social structure and be effective in our interactions.

FIGURE 26
But our basic mental-motor-perceptual machinery can't do much by itself.



And there's a whole subset of tools and methods that help us be effective in dealing with knowledge work. So take all those things together and call them an "augmentation system" (Fig. 27). That's what augments humans. And for good, practical purposes, let's divide the system into two parts. One part has all the technology in it and the other has all the rest. So I called these the "tool system" and "human system."

The *human system* includes training, knowledge, and skills that have to be installed as well as language, an extremely important invention that transcends, as an invention, anything else we have come up with. The methods, and all that we use to knit together all the little steps that we take during a day, are extremely important. Our customs for working, our procedures, the way our organizations are run, they're all done so that humans can realize effective results. Within this framework, any given capability that a human has is really a composite. The human capability is made up of the use of a lot of things and skills and training and conditioning, in addition to all the customs we just accept and the language that we've learned. It's this composite that we need to find ways to accentuate.

So along comes a lot of new technology for our tool system, which is great. But the technology side, by itself, is not sufficient. Our real capabilities are essentially hierarchical (Fig. 28). We learn a lot of lower-order capabilities, like writing and typing and reading. And we have built up many higher capabilities on top of these. So, if we bring in some new technology, like a longer lasting pen, it's going to make a little bit of effect. But if we bring in the kind of digital technology that was predictable, even in 1960, then potential changes throughout the whole system, affecting significantly the entire capability hierarchy, can begin to take place.

FIGURE 27

It is important to treat it as a two-part augmentation system; one part technology, the other the rest.

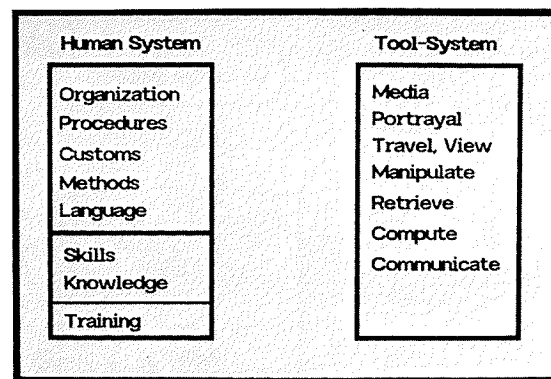
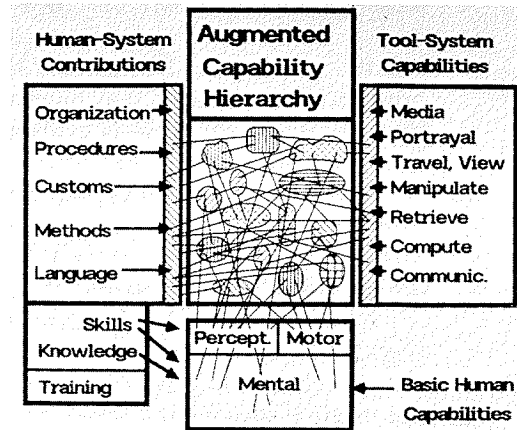


FIGURE 28
Capabilities grow
hierarchically.



It takes a long time (generations) to discover and implement all of the fruitful changes in the human system made possible by a given, radical improvement in technology. Where, as is the situation now, technology improves by rapid, large steps, it is predictable that the human system will become critically stressed in trying to adapt rapidly in ways that formerly took hundreds of years. There has to be a much-enhanced consciousness about concurrent evolution in the human system.

The technology side, the *tool system*, has inappropriately been driving the whole. What has to be established is a balanced coevolution between both parts. How do we establish an environment that yields this coevolution? Well, that's where the bootstrapping in a laboratory comes in. I said I wanted to do what I knew it was going to be like in our future. So we had to be more conscious of the candidates for change—in both the tool and the human systems. Whenever you hear somebody say it has to be "easy to learn and natural to use," put up a little flag and go question it. What's "natural"? Is there a natural way to guide a vehicle, as with reins? Well, that lasted a long time.

No! What's natural is what we have grown to accept. And if we look at how much learning it took to drive automobiles, and own and operate them, it would make ridiculous the things people say now regarding what they expect their computer system's learning to cost them. If it's going to be the kind of working companion that it's bound to be, then the learning part of it is relatively trivial. I know it's what sells now, because the market isn't very mature for people to buy things that look like they're going to be hard to learn. But I'm talking about the long-term trends, and the responsibilities of people that are doing the research and downstream planning. I think they should start

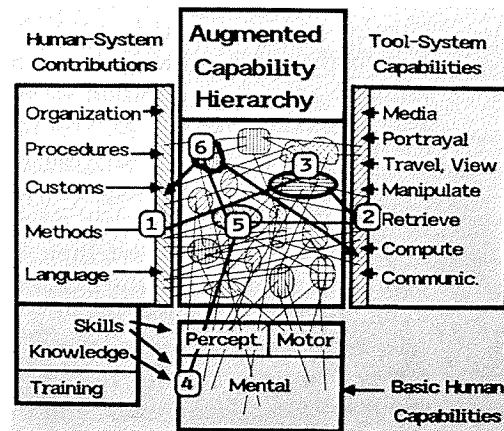
looking much more seriously for really significant gains in human performance and consider many more candidates for change, with special attention to the much-neglected human system candidates. This will bring up lots of things that might not have been even thought of before, because now there is explicit, active “prospecting” for ways to do things differently.

The mouse and the keyset came from trying to do something new in the tool system (similarly for structured files and their viewing and linking). I didn’t realize at the time how strange they’d seem and how long it would take for people to start considering them.

So then, how does the change take place? In a large organization (Fig. 29) there are lots of different parties taking part in a change. An idea for change in one place often “needs” help from another before the original idea can be implemented—for example, one might ask if the mail room could make one more daily delivery. Then, when such a need is fulfilled by a change in capability of one of the “lower” parts, it provides new “possibilities” for change among the “higher” parts that depend upon it. Evolution proceeds by reverberation: needs propagate downward, and when fulfilled, a new possibility propagates upward. Correspondingly, a possibility propagated by a new capability in a lower-level place can trigger a new idea in a higher place. Continuing the example, we can improve our operation by taking advantage of the extra mail delivery, but we’d need the copy center to change. . . . Possibilities tend to stimulate new needs.

So when we’ve got a fairly large organization, or large system of things, with specialists working all over the place that are responsible for changes in certain areas, there will have to be some extraordinary

FIGURE 29
Coevolution by reverberation—needs propagate downward and a new possibility propagates upward.



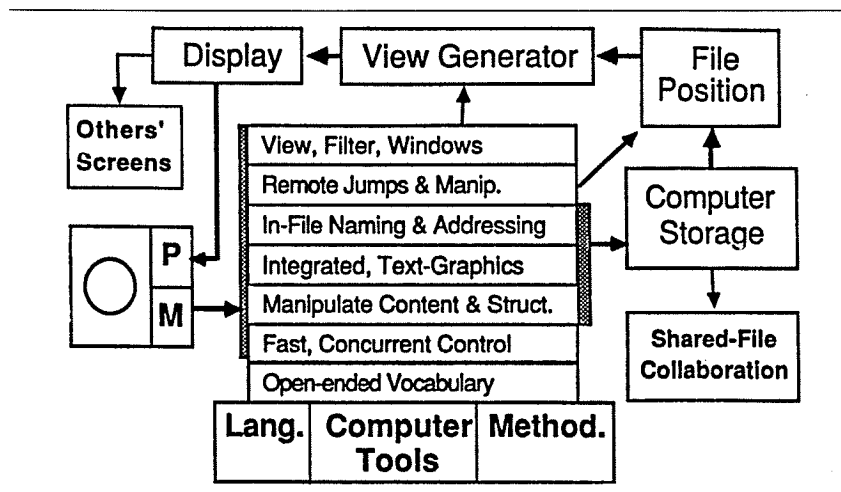
communicating to provide decade-frame evolution on a scale previously requiring centuries. Changes once took place so slowly that we weren't really aware of the evolutionary process. People moved and finally heard about something and tried it. But now, with a catapulting rate of change, and the pressures to become more effective, the changes within our organizations are just going to exceed the rate that we know how to deal with. Unless people start getting conscious and understanding about the evolution, for many, many structures this fast changing rate is going to exceed the elastic limit, and things are going to break.

So I said, "What better place could you start putting to work the new tools and things than in the process that is going to facilitate the evolution you have to go through." That produced a strong push toward computerized documents, and toward collaboration among groups.

One way to look at how to use the computer to help work with documents would be the possibility of a conventional word-processor approach. It's a very straightforward way. The orientation is to simulate paper on a display, targetted solely to produce hard copy. A considerable advantage in many situations, but it is a very anemic example of what the above framework promises.

It didn't even enter my mind to go with that picture. Instead, what my framework produced is shown in Fig. 30. The motor capability of the abstracted human module drives the computer tools and employs the associated methodological and conceptual-linguistic parts of his augmentation system in the special modes indicated. The result is quite a bit different from "word processing" and "office automation."

FIGURE 30



For instance, we should have an open-ended command vocabulary. Once we get hooked up to running in a more flexible environment, the computer tools must provide us with more and more functionality. Since we've got a very powerful beast that can do things, let's really look for fast control means where, for instance, we aren't limited to doing things sequentially, but can do things with both hands, concurrently. This stimulated the chord-keyset option.

I thought we should make it so that both content and structure can be stored and explicitly manipulated. Bring text and graphics together, because they're both important.

Every object in a file should be addressable, because I wanted to do remote jumping and manipulation. So we developed a very flexible addressing scheme that seems to fit well with a user's mental map of his working domain and offers handy addressing options that, in any command, can optionally be used in place of cursor selection. One should be able to move around in existing files rapidly and precisely, and easily jump across any intervening "file or content space" to an explicitly prescribed position. The computer should help navigate throughout the entire working space.

When I am positioned at a desired file object, I want to generate a view that will best serve my purpose of the moment. So, we evolved a flexible set of "view specifications" that can be invoked at any time—and an explicit opportunity is offered the user at any "jump" operation to designate "ViewSpecs." For example, "in this window, show only the first line of each statement, no blank lines between statements, show location addresses (e.g., "3b4") in front of each statement, show only the statements hierarchically "under" the targeted statement, and only those that are in the first two sub-levels." A content filter, of any degree of sophistication, can be invoked within any such ViewSpec so that, among the candidate statements already specified, only those that have a given content will be shown. Simple filters (e.g., to find a given embedded string) can be keyed in at any time and compiled on the fly. More sophisticated filter patterns can be stored as text strings that a user can specify with a "Set Content Pattern" command. Or, a programmer can write and compile very sophisticated filters that a user can easily designate by name to be instituted.

With a comprehensive addressing scheme it is easy to implement citation links, special text strings that both user and computer can interpret as addressing some object and also optionally specifying a particular view to be employed. "Jumping on a link" is a basic command, taking the user directly to the target object. A link can also be employed as part of the addressing in any other command where some file object is to be specified.

If I am moving around in somebody else's stuff, so I can study and analyze it much more effectively, I want to be able to train the

displays myself, because a display may not look as I expect it to look when I'm looking at it straight down on paper, and indeed it doesn't. And do I want it also so that I can share that display with others, so that we can collaborate? And share files? So that's the image, and that's why NLS was designed as it was.

ARCHITECTURE—TO SPECTRUM OF FUNCTION

Now to discuss architecture. I'll use a series of simple illustrations to lead up to the general approach we settled upon. Consider, as in Fig. 31, an application program on top of an operating system in a computer, serving a terminal. For any such application program, there are two facets: an interface process and the actual process that does the substantive work—two different parts. Let's think about them as two distinct but related design issues. For instance, I don't want the smart programmer who knows all about how this program works internally to think that he's the one to tell the world how to interface with it.

By 1968 we had begun evolving the programming language so that it was different for each part, and we could actually think and design for two separate modules (Fig. 32). The next step was to ask, Why, for each different application package, should you have a different front end? Frontends should be universal things, as in Fig. 33, to serve multiple (or all) applications for the user. So our language ideas were evolving to handle this approach. The system that we brought up on the SDS940 in 1968 was organized along these lines, as shown in Fig. 34. All the subroutines that did the application work were written with our special, MOL940 language that we had to develop ourselves. The control processes were specified in a control metalanguage, then compiled with a control-metalanguage translator into the control processor, which interacted with the user. We had a tree-meta translator that let

FIGURE 31
An application program has two different tasks: the frontend and the backend.

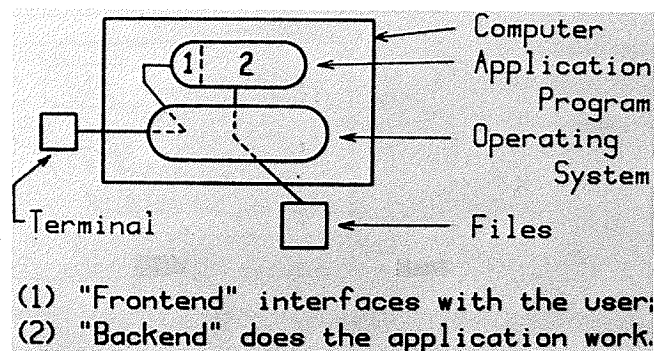
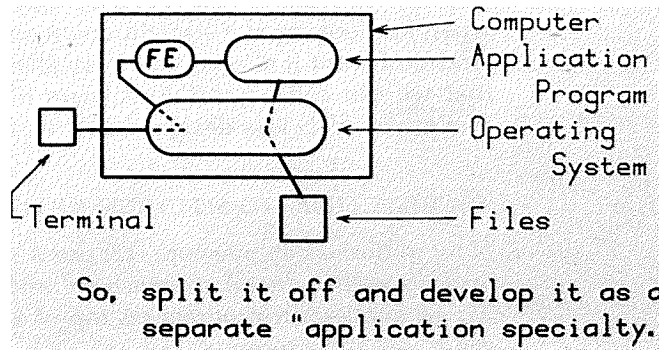


FIGURE 32
Frontend is actually a specialized application program divided into the interface part and actual process work.



us do our compiler compiling. We described a new compiler in a meta-compiler language, then compiled it with the tree-meta translator into the new, running compiler.

Incidentally, in those days I would talk about control language and control metalanguage rather than the command language because I felt that we were doing a lot of things that normally people don't think of as commands. But the pressure of external-world usage pulled us around so now we call it command language.

It was really fortunate for us to be involved in the ARPANET from the beginning. My early ideas of "community" support services could be made explicit in planning for the Network Information Center. One of the papers I wrote¹⁸ pointed out that besides allowing us to share data and process resources, the emergent networks will also provide us with a knowledge marketplace that's wide open for people sitting at their terminals all over the world. Topologically, Fig. 35 depicts what we assumed by the early 1970s would be the future environment for knowledge workers.

FIGURE 33
Why not provide a general purpose frontend, an all-application "User Interface System"?

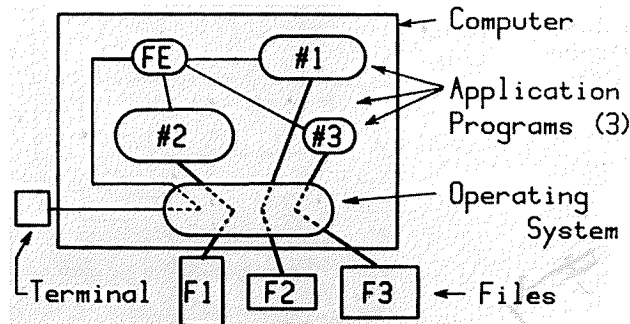
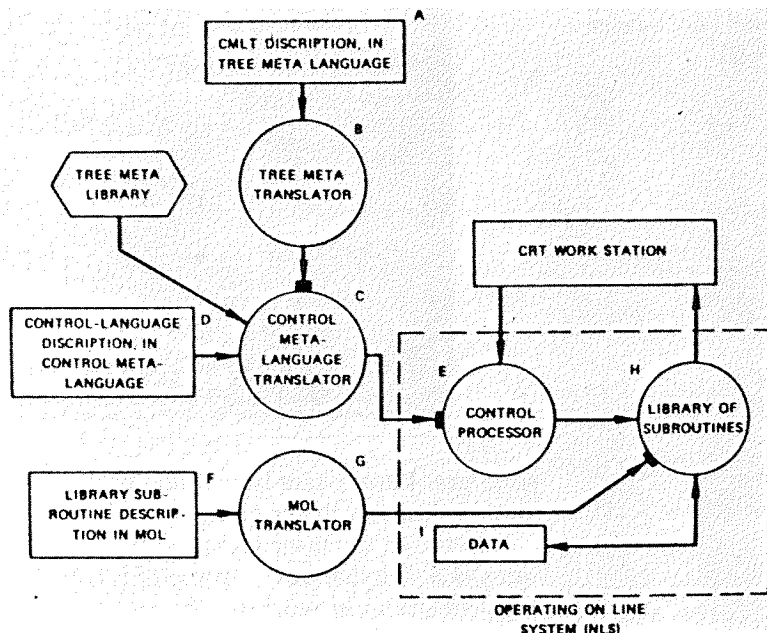


FIGURE 34
FJCC'68 system
architecture.



Let's not argue about how much functionality is in any one place. To the user, it doesn't matter whether the workstations are smart or not, as long as they do what he wants them to do. So let's look at topology. The topology that we wanted (Fig. 36), involved a user interface system (UIS) that's between the user's physical interface hardware (display, keyboard, mouse) and all the "smart" application

FIGURE 35
Our expected Tool
System architecture
(network of
networks).

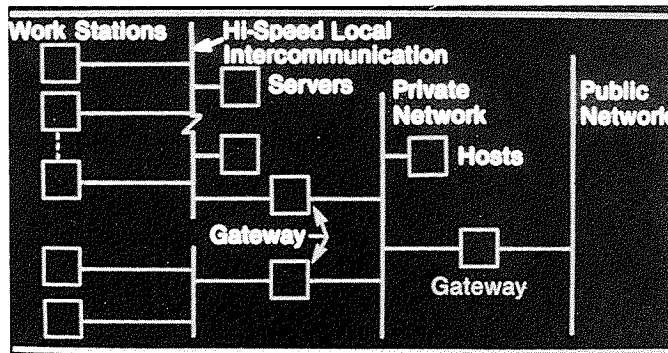
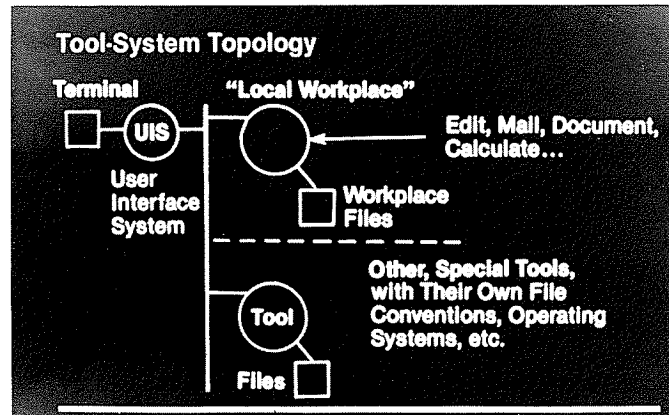


FIGURE 36
Desired Tool System
topology.



software. I don't care how much of these smarts are in his local workstation, but I'm never going to say that it all has to be there.

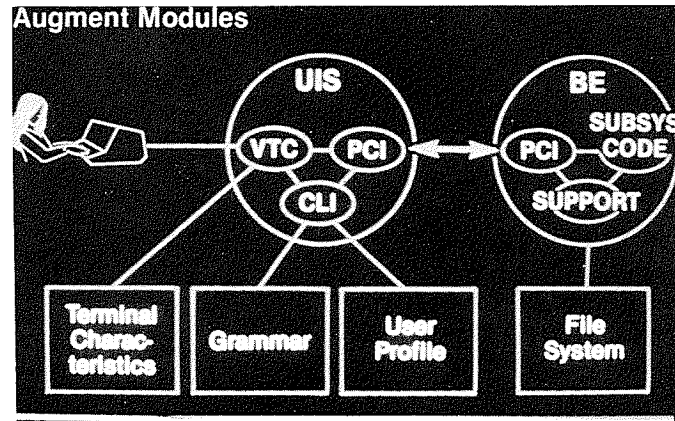
Between the user interface system and all of the application systems is a "virtual bus"—some UIS communication will be to applications in the local workstation, some to applications on the local-area network, and some may be far out through gateways to the UIS of a colleague. An important part of what we wanted to provide was a set of functions that the user thinks of as his private or "local" (topologically) workplace where he does his composing, studying, mail management, and calculations. But also he should be able to "reach through" his local workplace to get at all the other services.

The internal architecture of the UIS (Fig. 37) contains a virtual terminal controller (VTC) so all of the applications are programmed to deal with a standard, virtual terminal. The VTC translates to and from particular physical-terminal I/O streams, enabling different classes of terminal equipment for different classes of users—and also, importantly, makes evolution within the whole system easier.

The command language interpreter (CLI) interprets the user actions according to the command-language specification coded into the currently attached grammar file, which was created from a command metalanguage description via the command metalanguage compiler. The CLI also deals with a user-specific, user-profile file that enables individuals and classes of users to have independent control options.

The procedure call interface (PCI) modules translate back and forth between the procedure-call conventions of the various larger modules and the remote procedure call protocol we developed for use between arbitrary modules (including those connected via byte-sequential network circuits).

FIGURE 37
AUGMENT Modules
(User Interface
System with VTC,
CLI, PCI, and
grammar).



We called this architectural configuration "The Augmented Knowledge Workshop," and figured that all knowledge workers in the future would work in some such environment. The early framework concepts led us to believe that open-ended functionality was inevitable. Also, it emphasized how essential it was to facilitate the coevolution of the human and tool systems. One important objective in this architectural approach was to support this coevolution: Hardware and software could be changed with minimum disturbance to the human system and, conversely, changes in terminology, methodology, and functional dependence upon the tools could evolve with minimum disturbance to the tool system.

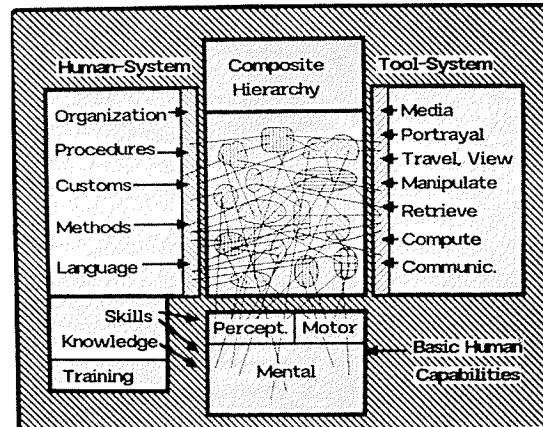
One way to illustrate payoff from this architecture is to consider the different profiles of functionality that different classes of users would likely want to employ as they look through their respective classes of terminals into their "knowledge workshops." A high-level project manager, a support clerk, or a skilled, heavy-working professional can each work with grammar and user-profile files that provide the functionality he or she needs, with command terminology and interaction modes suited to skill levels and ways of thinking.

All of this architecture was working by 1976, although it has only been recently that we could interest user organizations in trying to harness it toward an integrated knowledge-work environment. A number of our publications explicate various aspects of this architecture.¹⁹

Human Unit, Evolution, and Communities

It is important that the evolution of the user side should go on maximally; it has been badly neglected. The "basic human unit" is shown in Fig. 38. How are we going to evolve it? The reverberation concept

FIGURE 38
The basic human unit.



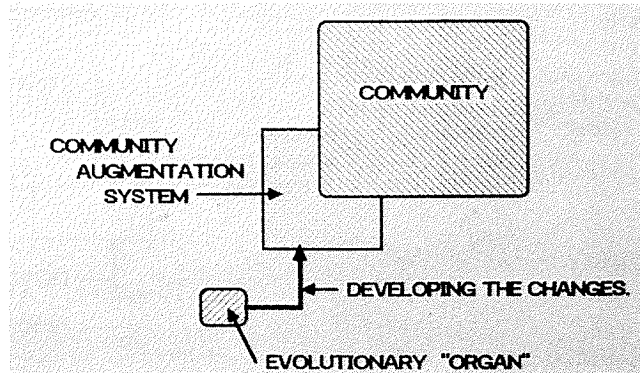
of its evolution is very important in my framework, and I have come to believe that it is only done by what you'd call "communities." Even in a large, highly structured organization much of the change process must involve stakeholders in groupings that are different from the line-management structure (i.e., like a community).

This is a big, important concept that started in the mid-1960s. If we depend critically upon a community that must interact in really effective collaboration, we need to build support systems especially designed for this purpose. Part of the conscious evolutionary process for our large organizations and institutions must therefore be to provide effective collaboration support for widely dispersed communities (Fig. 39).

But how was I going to promote an R&D program, with a "pilot community," to learn how to develop and provide effective "community support" tools and associated new collaborative methods? Well, in the spring of 1967, at an ARPA principal investigators meeting, Bob Taylor and Larry Roberts told us that ARPA was going to make computer network R&D a major program thrust, and that our "research computers" were going to be the ones connected by an experimental "ARPA Network." Even though many others were disturbed by the idea (because of perceived interference with their major research thrusts), I was thrilled. Finally, after listening to the initial interactions, I volunteered to develop and operate what became the ARPANET Network Information Center (NIC).

What I wanted was for the NIC to become a community support center that would really go after the development of collaboration-support tools and methods, and would provide services to encourage the ARPANET R&D folks to evolve their working ways accordingly.

FIGURE 39
An augmented community needs active, explicit, evolutionary mechanism!



Hopefully, there would emerge a subcommunity out there composed of those interested in the various aspects of augmenting (Fig. 40). Just consider the kind of leverage you get this way. So I've talked about a "bootstrap community" in my notes for over 15 years now, and it's something I still very much would like to see established (Fig. 41). Many features and capabilities built into NLS/AUGMENT were directly motivated by these community-support and distributed-collaboration concepts.

Unfortunately, the ARPANET grew so fast that the Network Information Center had to trim down what it could do functionally. We couldn't provide the extensive support services we had planned (discussed at some length in a 1972 paper²⁰), such as what we could do to support a community by integrating a lot of its dialog, and getting an intelligence system there; handbooks that evolved to integrate what to

FIGURE 40
Needed: A community for pursuit of maximum, whole-system capability!

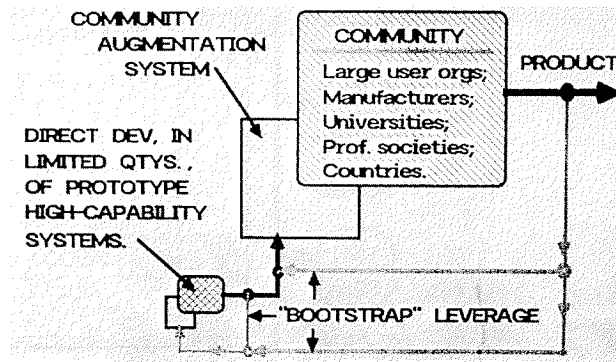
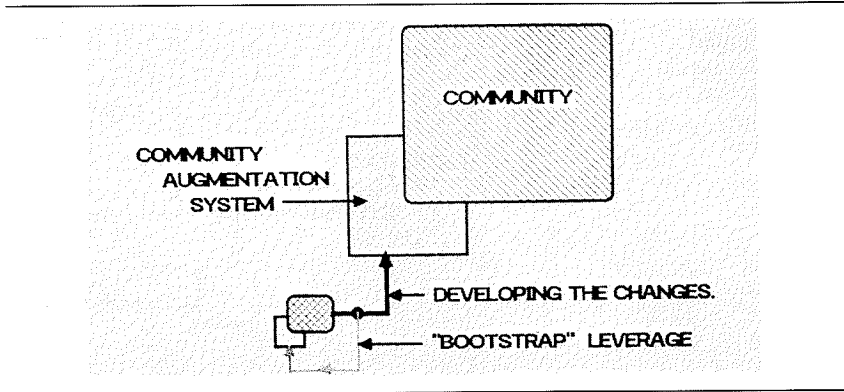


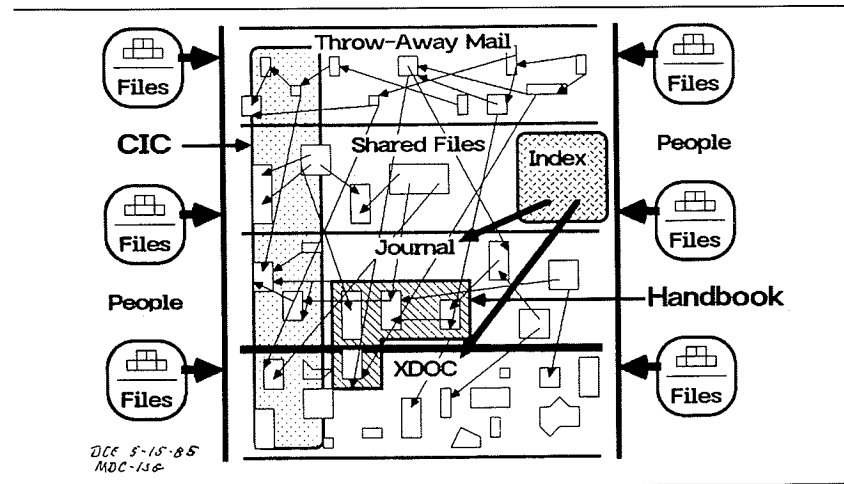
FIGURE 41
Strategy: Early augmentation system changes that also facilitate the evolution process.



do; and professional facilitation staff, which adds important value. There is a lot that can be embedded in this.

NLS/AUGMENT was built to be able to handle external document (XDOC) control, the community intelligence collection that everyone contributes to and participates in, and indexes to it. When the community has a special mission or disciplinary interest, we also could significantly facilitate the development and maintenance of a "dynamic community handbook" that integrates current status (terminology, hypotheses, conventions, plans, expectations, and so forth) in a coherent, self-consistent fashion. Figure 42 portrays all of this community support in a fashion peculiar to the NLS/AUGMENT set of features and capabilities. While each user had his own collection of working files, there would be a large collection of shared information—all of it

FIGURE 42
A community's handbook would be a periodically updated on/off-line publication.



embedded in uniform, composite-document structures. The generalized citation-link capability interconnects passages of any of these documents in meaningful ways as evolved with user conventions and collaborative methods. Ordinary, unrecorded mail and shared files gain significant value thereby. The Journal system provides its own unique value, considerably amplified with linkage use.

A "community intelligence collection" (CIC) can be distributed over all of the document types—embodying useful information and discussion about external activities relevant to the community's interests. Special indexing into this collection is again much enhanced with links. The Community Handbook fits into the picture nicely; any given "edition" is "published" in the Journal (and is probably available also in corresponding hard copy), which is controlled with the XDOC system.

I'm going to terminate at this point, since after 1976 we really had no chance to continue pursuing this "augmentation framework." It seemed no longer to fit in the pattern of the research at ARPA, or with what SRI wanted to do. When we landed out in the commercial world, we found it wasn't what people there wanted to do, either. The AUGMENT system stayed alive in sort of a funny, dumb way, often like taking a bulldozer in to help people work in their back yards. Then, McDonnell-Douglas bought Tymshare, and inside of its aerospace organization it's a very different situation because of the very heavy knowledge work involved. For a year I've been going out and talking with all those people involved in big projects and CAD systems, and finding out that these concepts are directly relevant to the needs and problems now being recognized there.

We're starting this year to build special interest communities. The first one is the AI community; possibly followed by the Ada™ community. Also, explicit consideration of integrated architectures similar to ours is under way, including not only virtual terminals, command interpreters, remote procedure calls, and shared-screen support, but also composite documents with addressable objects and citation links.

So that's a quick pass over my historical record.

REFERENCES

1. D. C. Engelbart, "Special Considerations of the Individual as a User, Generator, and Retriever of Information," *American Documentation* 12, No. 2, pp. 121-125, April 1961. (Paper presented at the Annual Meeting of the American Documentation Institute, Berkeley, Calif., 23-27 October 1960.)
2. J. L. Kennedy and G. H. Putt, "Administration of Research in a Research Corporation," RAND Corporation Report P-847, 20 April 1956.
3. D. C. Engelbart, "Augmenting Human Intellect: A Conceptual Framework," Summary Report, Stanford Research Institute, on Contract AF 49(638)-1024, October 1962, 134 pp.

4. D. C. Engelbart, "A Conceptual Framework for the Augmentation of Man's Intellect." In *Vistas in Information Handling*, edited by Howerton and Weeks, Washington, D.C.: Spartan Books, 1963, pp. 1-29.
5. W. K. English, D. C. Engelbart, and B. Huddart, "Computer-Aided Display Control," SRI Final Report to NASA Langley Research Center under Contract NAS 1-3988, July 1965, 104 p.
6. W. K. English, D. C. Engelbart, and M. L. Berman, "Display-Selection Techniques for Text Manipulation," *IEEE Transactions on Human Factors in Electronics*, Vol. HFE-8, No. 1, March 1967, pp. 5-15.
7. D. C. Engelbart and W. K. English, "A Research Center for Augmenting Human Intellect," *AFIPS Conference Proceedings*, Vol. 33, Fall Joint Computer Conference, San Francisco, December 1968, pp. 395-410.
8. See also D. C. Engelbart, "Computer-Augmented Management-System Research and Development of Augmentation Facility," Final Report, RADDC, 1970; and D. C. Engelbart, "Online Team Environment: Network Information Center and Computer Augmented Team Interaction," Final Report RADDC-TR-72-232 to Air Force Rome Air Development Center for SRI work under ARPA Order No. 967., 8 June 1972, 264 p.
9. D. C. Engelbart "Network Information Center and Computer Augmented Team Interaction," Technical Report RADDC-TR-71-175, as principal investigator for SRI Contract F30602-70-C-0219 under ARPA Order No. 967. June 1971, 99 p.
10. For descriptions see D. C. Engelbart, "Network Information Center"; and D. C. Engelbart, "NLS Teleconferencing Features: The Journal, and Shared-Screen Telephoning," *CompCon75*, 9-11 Sept. 1975, Digest of papers, pp. 173-176. (IEEE Catalog No. 75CH0988-6C.)
11. Described in D. C. Engelbart, "Toward High-Performance Knowledge Workers," *OAC'82 Digest, Proceedings of the AFIPS Office Automation Conference*, San Francisco, 5-7 April 1982, pp. 279-290.
12. See D. C. Engelbart, W. K. English, and J. F. Rulifson, "Development of a Multidisplay, Time-Shared Computer Facility and Computer-Augmented Management-System Research," SRI Final Report for AF Rome Air Development Center under Contract AF 30(602)-4103, April 1968, Engelbart, "NLS Teleconferencing"; and D. C. Engelbart, "Toward Integrated, Evolutionary Office Automation Systems," *Proceedings of the Joint Engineering Management Conference*, Denver, Colo., 16-18 October 1978, pp. 63-68.
13. Engelbart, "Toward High-Performance Knowledge Workers."
14. English, Engelbart, and Berman, "Display-Selection Techniques."
15. *Experimental Graphics Users' Guide*, SRI-ARC, 11 June 76, 32 pp.
16. *Output Processor Users' Guide*, SRI-ARC, 29 July 75, 97 pp.
17. Engelbart, "Augmenting Human Intellect."
18. D. C. Engelbart, "Coordinated Information Services for a Discipline- or Mission-Oriented Community," *Proceedings of the Second Annual Computer Communications Conference*, San Jose, Calif., 24 January 1972. (Also published in *Computer Communication Networks*, edited by R. L. Grimsdale and F. F. Kuo, Leyden: Noordhoff, 1975.
19. D. C. Engelbart, R. W. Watson, and J. C. Norton. "The Augmented Knowledge Workshop," *AFIPS Conference Proceedings*, Vol. 42, pp. 9-21, National Computer Conference, 4-8 June 1973; J. E. White, "A High-Level

- Framework for Network-Based Resource Sharing," *AFIPS Conference Proceedings*, June 1976, NCC, Vol. 45, pp. 561-570; R. W. Watson, "User Interface Design Issues for a Large Interactive System," *AFIPS Conference Proceedings*, Vol. 45, Montvale, N.J.: AFIPS Press, 1976, pp. 357-364; K. E. Victor, "The Design and Implementation of DAD, A Multiprocess, Multi-machine, Multilanguage Interactive Debugger," *Proceedings of the Tenth Hawaii International Conference on System Sciences*, University of Hawaii, 1977, pp. 196-199; Engelbart, "Toward Integrated"; and Engelbart, "Toward High-Performance Knowledge Workers."
20. Engelbart, "Coordinated Information Services."

OTHER IMPORTANT PUBLICATIONS

- Andrews, D. I., "Line Processor—A device for amplification of display terminal capabilities for text manipulation," *AFIPS Conference Proceedings*, pp. 257-265, National Computer Conference, 1974.
- "Augmenting Your Intellect," editorial interview with D. C. Engelbart, *Research/Development*, August 1968, pp. 22-27.
- Bush, V., "As We May Think," *The Atlantic Monthly*, July 1945, pp. 101-108. (See also this volume.)
- Culler, G. J., and R. W. Huff, "Solution on Non-Linear Integral Equations Using On-Line Computer Control," paper prepared for Spring Joint Computer Conference, San Francisco, May 1962. (Assumedly published in *Proceedings Spring Joint Computer Conference*, Vol. 21, [Palo Alto, Calif.: National Press, May 1962].)
- Engelbart, D. C., "Augmenting Human Intellect: Experiments, Concepts, and Possibilities," Summary Report, Stanford Research Institute, under Contract AF 49(638)-1024 for Directorate of Information Sciences, Air Force Office of Scientific Research, March 1965. SRI Project 3578; 65 p.
- , "Intellectual Implications of Multi-Access Computer Networks," *Proceedings of the Interdisciplinary Conference on Multi-Access Computer Networks*, Austin, Texas, April 1970.
- , "Design Considerations for Knowledge Workshop Terminals," *AFIPS Conference Proceedings*, Vol. 42, pp. 221-227, National Computer Conference, 4-8 June 1973.
- , "Evolving the Organization of the Future: A Point of View." In "Emerging Office Systems," *Proceedings of the Stanford International Symposium on Office Automation*, March 23-25, 1980, edited by R. Landau, J. Bair, and J. Siegman, Norwood, N.J.: Ablex Publications Corporation.
- , "Collaboration Support Provisions in AUGMENT," *OAC '84 Digest, Proceedings of the 1984 AFIPS Office Automation Conference*, Los Angeles, Calif., February 20-22, pp. 51-58.
- , "Authorship Provisions in AUGMENT," *COMPCON '84 Digest, Proceedings of the 1984 COMPCON Conference*, San Francisco, Calif., February 27-March 1, 1984, pp. 465-472.
- Engelbart, D. C., and B. Huddart, "Research on Computer-Augmented Information Management," Technical Documentary Report No. ESD-TDR-65-168, under Contract No. AF 19 (628)-4088 from Electronic Systems Division, Air Force Systems Command, USAF, March, 1965. 128 pp.

- Foth, T., "The Origin, Anatomy, and Varieties of Mus Computeralis" (It's the Year of the Mouse!), *Softalk*, Vol. 2, April 1984, pp. 88-96.
- Levy, S., "Of mice and Men," *Popular Computing*, May 1984, pp. 70, 75-78.
- Licklider, J. C. R., "Man-Computer Symbiosis," *IRE Transactions on Human Factors in Electronics*, March 1960.
- Licklider, J. C. R., and W. E. Clark, "On-Line Man-Computer Communication," *Proceedings of the Spring Joint Computer Conference*, Vol. 21, pp. 113-128 (Palo Alto, Calif.: National Press, May 1962).
- Licklider, J. C. R., and Robert W. Taylor, with Evan Herbert editor, "The Computer as a Communication Device," *Science & Technology*, April 1968, pp. 21-31.
- Lindgren, N., "Toward the Decentralized Intellectual," *Innovation*, No. 24, September 1971.
- Rheingold, H., "The Loneliness of a Long-Distance Thinker," chapter nine, pp. 174-204, in *Tools for Thought: The People and Ideas behind the Next Computer Revolution*, New York: Simon & Schuster, 1985. 335 pp.
- Victor, K. E., "A Software Engineering Environment," *Proceedings of AIAA/NASA/IEEE/ACM Computers In Aerospace Conference*, Los Angeles, Calif., October 31-November 2, 1977, pp. 399-403.

Participants Discussion

Hank Strub

I'd like to hear your feelings on why it took so long for the mouse to be adopted into personal workstation technology, and why that hasn't happened yet with the chord keyset?

Engelbart

It's part of a larger story that I think fits in with culture. If you looked at the whole map I drew you would see that on the human system side all the opportunities have changed; the technology side has grown way out of proportion, in my view, just stupendously so. Fifth generation and all the hoopla about it is just all the more pushed in there, so there's got to be a balance. There's a reason that in our culture we grew up and absorbed all of the human system side that we used without even questioning it or thinking of it—they didn't come to us as inventions. We look upon other cultures where there is not the idea of "progress," and we laugh at them. But all we have to do is turn and look at ourselves to realize that our culture has not yet understood that the human side is open for progress and change. We more or less resist the change and say, "Boy, I'll be damned if I'm going to let that computer make me change; if it's so smart it ought to do it for me." We're totally missing the point, in my estimation. Well, people look at the mouse and say, "Geez, who wants to do all that?" Or they say the keyset is even worse. It's part of the culture.

The biggest challenge that's ahead for us, if we want to make progress, is to start affecting the perceptions that people have about what potential is there. We need to start looking for change and finding out a rational way to do the evolution so it doesn't break us apart. I just love to get into a dialog with people about the strategies for that evolution. I found people at McDonnell Douglas who were talking about it. There are top level guys who say, "We're going to remake the corporation." That's given people at many of the other levels the courage to go ahead and try. I come and say, strategically, we ought to do this; so they're sticking their necks out and giving me a chance to start.

Greg Heil

I'd like to ask you, have you considered combining the mouse and the keyset into a single device, a handwriting recognition system?

Engelbart

Yes, I thought I'd have a mouse and a keyset in each hand. There's a lot of potential to that, and people say, "Hey, what about this number of buttons on a keyset? Why did you pick three buttons on your mouse?" Why, it was all we could get in at the time. So, yes, I'd go

for more. There's a lot to learn. I did a lot of thinking one time about the ways you could get transducers to pick up almost any kind of a signal that your motor system or nerve system could produce and translate that into actions you want; it's a whole new control language you could learn. I was just wondering about the bandwidth and the effect of that, and I could just picture really going through space—it's wide open.

Greg Heil

My suggestion was not to increase the number of transducers, but to decrease them to a single one, to a handwriting recognizer, so that you don't have extra buttons; you only use one hand to do it.

Engelbart

Well, that's a pretty slow way to enter characters, at least for English. Try it, but I think that if you've got such a marvelous machine there, you don't want to keep driving it with reins. And that's what I think would happen with the handwriting recognizer. I feel similarly about speech recognition. It's very nice in a lot of ways, but you couldn't control a system like we had with speech commands; you couldn't go fast enough unless you got some very sophisticated signals.

Charles Irby

I have two questions, one is on the tool system side and the other is on the human system side. Let's take the human system side first: Given the work that you've done in this area, what insights do you have about the changes that are going to be required in the human system, given the changes in technology in the tool side that we are experiencing today?

Engelbart

Well, I've got a lot of general ideas and I can tell you some of those. The main need is to find an evolutionary process that starts the exploration. Many people and many things to try. An example is that our organizations can get much flatter. The kind of communication can be so much greater, that if I could just go like this, and be in voice contact with you, same screens, diving around the information we're familiar with, you can be in any place in the country. It can change considerably the kind of roles that we can have, lots of specialty roles that can flip in and out of your work to help you from any place in the country. There isn't any way it would be practical with the kind of time it would take for you to get together and look at your stuff. Administrative and managerial structures can be different, and things like the matrix organizations, which have a lot of appeal in project-oriented environment but are difficult to administer, have a whole new breath of life coming to them because of the kind of coordination communication. Working groups that can be separated geographically. We do it a lot of ways now, but no one has been really focusing on the pursuit of ways to collaborate.

Charles Irby

On the tools side, a lot of the technology that was developed at SRI has readily transferred into other commercial products. Yet there are some, such as the structured text and the view controls that you had over the structured text, the linking mechanism from one place in the document to another place in possibly another document, and the notion of a mail system that is supported by a database, automatically cataloging all of the mail that has been sent. These things haven't readily transferred into other products. Why is that?

Engelbart

I was hoping the historians would answer that. I don't know, it's puzzled me. I used to take it personally and every once in a while it flashes . . . all I can say is that it just didn't fit cultural perceptions. When the computers first started becoming really available—personal computers or time-sharing—it was enough for people to make the adjustment to start interacting with them. If the marketplace isn't ready for something like that, who's going to invest money in trying to get it out there? Who has to decide to invest the money? It's guys who have been in the business for a while, and they have to depend upon consultants. They're experienced and familiar with things that aren't very often the ones that are in the vanguard. The whole process of who gets it out there and decides they are going to invest the money in it and risks it or goes out there and tries to train the customers. Once you get a system built, then you've got to put a lot of money into PR that tells everybody that's the best. So, you're doing yourself in down the way if, later on, you come back and say, now this is the best. So anyway, a big part of the user organizations have to get in gear and start saying, "Quit coddling us, we're ready to change. We're looking for what to do better, don't give us this bull about what's easy to learn."

Larry Press

I notice in reading one of your early reports that you paid a lot of respect, I guess, and commented a lot on Vannevar Bush's suggestions. And I just wondered if you could comment on how important you feel he was, his thinking and also his work, to the development of all this interactive computing.

Engelbart

I don't know about other people in that respect. For me, it was part of a singular thing, because I was a little navy boy, an electronics technician in World War II out in the Philippines, and getting moved from one place to another. They stuck you on an island to wait to get you assigned to somebody else. Somebody said there was a library there. It was a Red Cross library, up on stilts in a native hut, really neat, nobody there. I was poking around and found this article in *Life* magazine about his memex, and it just thrilled the hell out of me that people were thinking about something like that. So I didn't really act on that, but I'm sure later, as I got into this, that it started to affect me. Later,

when we were starting the parts of the way NLS was built, there's a thing called a sequence generator that you could flip in your own user choices of a sequence that goes and picks you this thing and that; the links were a lot coming from Bush's idea of trail generation, through his documents. So it was a very explicit connection in that respect. I wish I could have met him, but by the time I caught on to the work, he was already in a nursing home and wasn't available. History is moving faster nowadays, incidentally, so you can find some of us old fossils that are still alive.

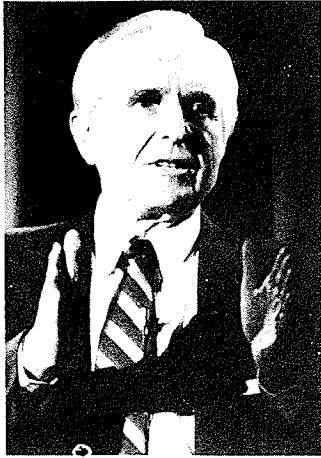
SESSION 4: 3:30–5:30

THE AUGMENTED KNOWLEDGE WORKSHOP

Doug Engelbart
Introduction by Charles Irby

When I left Glen's group, I spent a year working for Lytton Industries on the ground control system supporting a precursor to skylab. Then I went to a conference that was being held up in San Francisco. It was the Fall Joint Computer Conference. This particular conference was very, very special for me and for a lot of people because there was an hour and a half session in which Douglas Engelbart from the Augmentation Research Center at SRI in Menlo Park, California, gave a live video demonstration of a highly interactive computer graphics and computer text manipulation system that had been developed at SRI. I went to that particular session not knowing what to expect, and I was completely blown away. I happened to find afterward the particular person who seemed to be in charge technically; his name was Bill English. I cornered him and said, "This is really nifty and I think I can help you." And he said, "We're looking for a few good men. Why don't you come by?" The next week I went by and the personnel department said, "No, we have no openings." I said, "Wrong. I'm going to sit here until Bill English comes to talk to me." (I was fairly bold in those days.) And he did. They hired me, and it was the beginning of a seven-year effort on my part at SRI with Doug Engelbart. A great many of the concepts that people now think of as commonplace, like the mouse and multiple windows, editing across windows, integrated text and graphics—all those things originated, from my point of view, from Doug and his group at SRI.

I'd like to introduce Doug Engelbart and his perspective of the last 20 years or so.



Doug Engelbart is a pioneer in the design of modern interactive computer environments. He holds the patent on the mouse; created the first two-dimensional editing system, and was the first to demonstrate such things as the use of remote procedure protocols, mixed text-graphics, and shared screen viewing.

Doug holds a B.S. in EE from Oregon State University and a Ph.D. in Electrical Engineering from the University of California at Berkeley. His career has led him from the U.S. Navy as an electronics technician to NASA's Ames Research Laboratory and then to Stanford Research Institute (now SRI International) where he led the team that designed and built the NLS "Augmented Knowledge Workshop."

From January 1977 to March 1984 Doug was a Senior Scientist at Tymshare, Inc., Cupertino, California. In 1984, Tymshare was acquired by McDonnell Douglas. Doug continues as a Senior Scientist in the Information Systems Group, promoting the type of integrated-system architecture conceived and implemented by him at SRI International.

The Augmented Knowledge Workshop

Doug Engelbart

McDonnell Douglas

Introduction

The story of my involvement with on-line workstations begins in early 1951 with a vision and a life-time professional commitment. Over 34 years of pursuit have created a lot of personal history, and the object of this historical exercise, the workstation, occupies a unique place in it.

For me, a workstation is the portal into a person's "augmented knowledge workshop"—the place in which he finds the data and tools with which he does his knowledge work, and through which he collaborates with similarly equipped workers. I consider that the large system of concepts, skills, knowledge, and methods on the human side of the workstation has to be taken into account, in a balanced way, when pursuing increased human effectiveness. So, my workstation-history story embraces a rather large sphere.

The task of writing an historical piece is unfamiliar enough to cause me difficulty by itself, but the associated stirring of old records and old memories has added a nearly overwhelming burden of dreams, events, people, stresses, pleasures, disappointments—the firsts and the failures. Now, what from all of this—and how to organize it—will make an appropriate "history" paper? I could provide a solid measure of objective reporting—events and dates. I have been an involved observer of related computer history since 1951. I watched and experienced how the supportive hardware, languages, and architecture evolved; witnessed the people and efforts that brought time-sharing into being; and was even more closely involved with the emergence of computer networks. Through all of this, I was wholly focused on what these things could do for people at workstations. And then there was office automation and personal computers; you don't have to be an old guy to have watched these emerge, but I'm sure they looked different to me than to most.

I could also provide lots of objective reporting about the events and dates associated with the things I have caused or had a direct hand in. There seems to be a lot there that is quite relevant to this "history of the workstation" theme. It was dusty, laborious work, this process of brainstorming for candidates, culling and ordering and trying to describe them in some reasonable sequence and context. But what I came

to realize is that there is one, clearly dominant factor that underlies essentially every cause for any uniqueness that I might list for historical record. It isn't a technology, it isn't a science, and it isn't a marketing or business model. And I am going to give it dominant coverage in this paper. It is what I call my "Framework." My Framework is based upon an intuitive conviction, implanted in my head (apparently permanently) over 30 years ago, that the gains in human knowledge-work capability that we will achieve by properly harnessing this new technology will be very large. Metaphorically, I see the augmented organization or institution of the future as changing, not as an organism merely to be a bigger and faster snail, but to achieve such new levels of sensory capability, speed, power, and coordination as to become a new species—a cat.

Based upon this conviction about huge potential gains for mankind, my Framework explains for me generally where such gains are going to come from, and provides strategic principles that can help guide a conscious pursuit of these gains.

Genesis

I was several years out of school, possessing a B.S. in electrical engineering and two years' experience during World War II (halfway through college) as an electronics technician. I was doing odd-job electrical engineering work at Ames Research Laboratory in Mountain View, California, with the National Advisory Committee for Aeronautics (NACA, forerunner of NASA). For several months I had been devoting most of my spare time to searching for professional goals; for some reason I wanted to invest the rest of my heretofore aimless career toward making the most difference in improving the lot of the human race.

I had initially dashed off in many fanciful directions, but yet managed enough interludes of reasonably sober thinking to build up some useful, strategic generalizations. Retreading myself professionally, to become proficient and then extraordinarily productive in some new field wasn't worth considering without a significantly attractive scenario, embedded in a reasonably structured strategic framework. The high-payoff scenarios all seemed to involve creating or joining something that, however disguised, would essentially be a crusade. Crusades have many strikes against them at the outset. In particular, they don't connect to a normal source of government or business revenue. They don't have nice organizational frameworks. You can't go out on the streets and expect to find financial, production, or marketing vice-presidents interested in the crusade. Moreover, even if you accomplish the sweeping change that is the ultimate objective, chances are that in

this very complex world the side effects might be bad enough to make you wish you hadn't tried.

Suddenly, up through all of this delightful, youthful abstraction bobbed the following clear realization: The complexity of the human situation was steadily increasing; not only that, but its rate of increase was increasing. Along with the increasing complexity had come a general increase in the urgency associated with the more critical problems. If one invented a measure for each of these—complexity and urgency—then for a given problem, the product of its complexity times its urgency would represent a fair measure of the difficulty mankind would find in dealing with that problem.

- FLASH-1: The difficulty of mankind's problems was increasing at a greater rate than our ability to cope. (We are in trouble.)
- FLASH-2: Boosting mankind's ability to deal with complex, urgent problems would be an attractive candidate as an arena in which a young person might try to "make the most difference." (Yes, but there's that question of what does the young electrical engineer do about it? Retread for a role as educator, research psychologist, legislator, . . . ? Is there any handle there that an electrical engineer could . . . ?)
- FLASH-3: Ahah—graphic vision surges forth of me sitting at a large CRT console, working in ways that are rapidly evolving in front of my eyes (beginning from memories of the radar-screen consoles I used to service).

The imagery of FLASH-3 evolved within a few days to a general information environment where the basic concept was a document that would include mixed text and graphic portrayals on the CRT. The imagery carried on to extensions of the symbology and methodology that we humans could employ to do our heavy thinking. There were also images of other people at consoles attached to the same computer complex, simultaneously working in a collaboration mode that would be much closer and more effective than we had ever been able to accomplish.

Within weeks I had committed my career to "augmenting the human intellect." In a few months, I left the NACA and enrolled as a graduate student at UC Berkeley, where Professor Paul Morton had started a computer science activity (although it would be many years before universities began calling it that). He was several years along in developing the California Digital Computer (CALDIC).

Within a few years I had to accept the fact that research on any kind of interactive computer applications would not provide me with a program acceptable to the university community for Ph.D. and later

faculty pursuit. So, I settled for something else, got my Ph.D., and went to Stanford Research Institute (SRI) where I hoped ultimately to promote support for an augmentation program.

FRAMEWORK

That was 1957. By 1959 I was lucky enough to get a small grant from the Air Force Office of Scientific Research (AFOSR, from Harold Wooster and Rowena Swanson) that carried me for several years—not enough for my full-time work, but by 1960 SRI began pitching in the difference.

It was remarkably slow and sweaty work. I first tried to find close relevance within established disciplines. For a while I thought that the emergent AI field might provide me with an overlap of mutual interest. But in each case I found that the people I would talk with would immediately translate my admittedly strange (for the times) statements of purpose and possibility into their own discipline's framework. When rephrased and discussed from those other perceptions, the "augmentation" pictures were remarkably pallid and limited compared to the images that were driving me.

For example, I gave a paper in 1960 at the annual meeting of the American Documentation Institute, outlining the probable effects of future personal-support use of computers. I discussed how a focus on personal support would change the role of their future systems and how such a change would enable more effective roles for the documentation and information specialists.¹ I received no response at all at the meeting. One reviewer gave a very ho-hum description of the paper as the discussion of a (yet another) personal retrieval system. Later, during a visit to a high-caliber research outfit, an information-retrieval researcher got very hot under the collar because I wouldn't accept his perception that all that the personal-use augmentation support I was projecting amounted to, pure and simple, was a matter of information retrieval—and why didn't I just join their forefront problem pursuits and stop setting myself apart?

Then I discovered a great little RAND report written by Kennedy and Putt² that described my situation marvelously and recommended a solution. Their thesis was that when launching a project of inter- or new-discipline nature, the researcher would encounter consistent problems in approaching people in established disciplines. They wouldn't perceive your formulations and goals as relevant, and they would become disputative on the apparent basis that your positions were contrary to "accepted" knowledge or methods. The trouble, said these authors, was that each established discipline has its own "conceptual framework." The enculturation of young professionals with their discipline's framework begins in their first year of professional

school. Without such a framework, tailored for the goals, values, and general environment of the respective discipline, there could be no effective, collaborative work. Furthermore, if such a conceptual framework did not already exist for a new type of research, then before effective research should be attempted, an appropriate, unique framework needs to be created. They called this framework-creation process the "Search Phase."

So, I realized that I had to develop an appropriate conceptual framework for the augmentation pursuit that I was hooked on. That search phase was not only very sweaty, but very lonely. In 1962, I published an SRI report entitled, "Augmenting Human Intellect: A Conceptual Framework."³ With the considerable help of Rowena Swanson, this was condensed into a chapter of a book published in 1963.⁴ I can appreciate that these framework documents appear to many others as unusably general and vague. But, for me, the concepts, principles, and strategies embodied in that framework look better and better every year. The genesis of most of what was and is unique about the products of the augmentation work can be traced back to this framework, which I'll return to later with a fuller description.

PROGRAM SUPPORT

I submitted many proposals before getting support to pursue the augmentation program outlined in the Framework report. Among the stream of politely phrased regrets, there was one that, in contrast to today's environment, can provide useful perspective on the environment of 1961. Four high-quality civilian experts had been enlisted by one agency as a site-visit team—a brain researcher, a psychologist, and a computer expert—and for me it was a very enjoyable day's dialog. But the subsequent letter from the agency informed me regretfully that (paraphrased) ". . . since your interesting research would require exceptionally advanced programming support, and since your Palo Alto area is so far from the centers of computer expertise, we don't think that you could staff your project adequately. . . ."

When J. C. R. Licklider ("Lick") came from Cambridge to take over ARPA's newly formed Information Processing Techniques Office (IPTO) in late 1962, I was figuratively standing at the door with the "Conceptual Framework" report and a proposal. There the unlucky fellow was, having advertised that "man-computer symbiosis," computer time-sharing, and man-computer interfaces were the new directions. How could he in reasonable consistency turn this down, even if it was way out there in Menlo Park?

Lick moved very swiftly. By early 1963 we had a funded project. But, whereas I had proposed using a local computer and building an interactive workstation, Lick asked us instead to connect a display to

the System Development Corporation's (SDC's) AN/FSQ32 computer, on site in Santa Monica, to do our experimenting under the Q32's projected new time-sharing system. (Converting the Q32 to be a time-shared machine was SDC's IPTO project.)

Later that year, our project was modified to include an online data link from Menlo Park to Santa Monica, with a CDC 160A minicomputer at our end for a communication manager, supporting our small-display workstation. For various reasons, not uncommon in pioneering ventures, that first year was very unproductive relative to the purposes and plan of our project. Lick was willing to put some more support into the direct goal (more or less as originally proposed), but the support level he could offer wasn't enough to pay for both a small research staff and some interactive computer support. Mind you, the CDC 160A, which was the only commercially suitable minicomputer that we knew of, even though having only 8K of 12-bit words, and running at about 6 microseconds per instruction, cost well over \$100,000 (1963 dollars). It had paper tape in and out; if the system crashed, you had to load the application program from paper tape, and the most recent dump of your working file (paper tape), before you could continue. A crude, industry-standard Flexowriter (online typewriter) could be driven; otherwise it was paper tape in and out.

What saved my program from extinction then was the arrival of an out-of-the-blue support offer from Bob Taylor, who at that time was a psychologist working at NASA headquarters. I had visited him months before, leaving copies of the Framework report and our proposal. I had been unaware that meanwhile he had been seeking funds and a contracting channel to provide some support. The combined ARPA and NASA support enabled us to equip ourselves and begin developing Version 1 of what evolved into the NLS and AUGMENT systems. Paul Fuhrmeister, and later Eugene Gribble of NASA's Langley Research Center, had to stick out their necks as successive heads of Langley's large computational division to support the direction and supervise NASA's support for our program, which continued several years after Taylor left NASA to join ARPA's IPTO office.

Our ARPA support grew and was fostered by Lick's successors—Ivan Sutherland, Bob Taylor, and Larry Roberts. Meanwhile, the Air Force's Rome Air Development Center, at Rome, New York, began to supply supporting funds. By 1967, it was recognized that the respective contributions from ARPA, NASA, and RADC represented significant parts of a coordinated program. The other agencies began funneling their funds through RADC, which served for many years to both monitor and manage our contracts, as well as provide their own significant share of support funds. John McNamara and Duane Stone provided strong support and contract liaison from RADC.

NASA support ended by 1969, and ARPA and RADC provided

significant support until 1977, although from 1974 the funding became even more for supporting applications and developments for other organizations, for targets formulated by others (e.g., the National Software Works). The continuing pursuit of augmentation along my strategic vector virtually stopped.

The Augmentation Research Center (ARC)

An historically important organizational cluster emerged at Stanford Research Institute in the 1960s, peaked about 1974, and was scattered in 1977—with a small core carrying forth in a commercial (and then industrial) environment to the present. It grew by ones and twos from 1963, as it collected “permanent” members from the SRI technical staff, and recruited new ones from the outside. By 1969 I believe we were about 18 strong; this grew steadily until by 1976 we totalled about 45. In 1973 we made two explicit subgroups, one headed by Dick Watson doing development of software (and some hardware), and one headed by Jim Norton handling operations and applications support.

SRI was organized by divisions, each containing a group of laboratories, the hierarchy being formed according to the associated disciplines. ARC grew to laboratory size and status, but it became something of a problem for SRI. Other laboratories (at least in science and engineering) operated more or less as a “farmers’ market,” where small and changing clusters of researchers promoted and conducted research projects as a loose federation. The management structure, budgeting, accounting, and financing for the Institute had evolved to support this kind of business. But ARC was driven by a coherent, long-term pursuit. This involved the continuing evolution of an ever-larger and more sophisticated system of hardware and software. It also came to involve delivering solid support service to outside clients to provide meaningful environments for learning about the all-important process of coevolution between the human-system and tool-system components of our organizations (as per my conceptual framework).

It didn’t seem unreasonable to me to pursue this course; things similar and on a grander scale are common for other researchers. It is taken for granted, for example, that funding agencies will build and operate accelerators and observatories in support of research in nuclear physics and astronomy, or will outfit ships and airplanes to support research expeditions. But whatever my perception, there were some significant problems and stresses with which our over-all environment didn’t have effective ways to cope. In the particular dynamics involved, there were probably seven relevant parties: me, the ARC staff, other SRI researchers, SRI management and administration, ARC’s sponsors, ARC’s utility-service clients, and other groups of researchers outside of SRI. It would be an interesting historical study to try to un-

derstand the diversity of perception that must have existed among this set of players. What did the different parties perceive for the future of workstations, for the range of function and application that would come about, for the systems architectures and standards that must emerge, and for the impact on the organizations that learned how to harness these most successfully?

Even as a central party in what happened, I've not understood the dynamics. But I am pretty sure that disparities among the perceptions of all of the above parties had a major part in what to me was the "great collapse of SRI-ARC." Even if I had done everything right over the years (a laughable hypothesis), it is now fairly clear to me that it isn't the market's fault if someone fails in trying to sell it something that the market isn't ready for. In other words, I can't blame those other groups. (Which of course makes for a personal problem, since during those times of black discouragement when one wants desperately to blame someone, there is only one candidate—that guy at the head of the list.)

In 1977, SRI judged it better to move our large-system development and external-service activities out from the research institute environment and into a suitable commercial environment. They advertised, entertained prospective bidders, made a selection, and negotiated a transfer of the business to Tymshare, Inc., of Cupertino, California. The system was renamed AUGMENT and marketed as part of Tymshare's integrated Office Automation services. In 1984, McDonnell Douglas Corporation acquired Tymshare, and the small AUGMENT business is now operated as the Augmentation Systems Division of the Computer Systems Company within the MDC Information Systems Group.

SOME OF ARC'S EARLY PRODUCTS—1964 TO 1968

Let's take a look at some of the actual experiments we tried during the years from 1964 to 1968 at SRI. Around 1964 we got off the CDC-160A system, which in fact we were using as a personal system. This was our first, real, stand-alone machine.

Screen Selection Tests and the Mouse

In those days the cost of getting display systems to work was very high. My strategic plan under the above-mentioned framework was to skip the then-prevalent, interactive typewriters and focus from the outset on displays. My assumption was that there is a great deal to learn about how to harness highly interactive display capability, and by the time we really learned how, the prices would be down considerably.

We wanted to start early experimenting with screen selection. The idea of working and interacting very actively with the display meant

that we had to tell the computer what we were looking at, so we needed a screen selection device. There was a lot of argument about light pens and tracking balls in those days, but none of those arguments served our needs very directly. I wanted to find the best thing that would serve us in the context in which we wanted to work—text and structured items and interactive commands.

The context was important. So we set up computer-controlled experiments oriented to our working mode, where we assumed that purposeful knowledge workers would be spending a significant portion of their time writing, studying, analyzing, and even debugging. We collected a set of candidate screen-selection devices to test. We did our experiments with our one workstation of that period, which is shown in Fig. 1, together with one of our test devices. In trying to be complete about the array of test devices, I dug up some old notes of mine describing a possibility that eventually turned into the very first mouse (Fig. 2). The tests were carefully run, and we even integrated selection errors and their correction penalties in evaluating the "goodness" of a device. The experiments and their results were fully reported,⁵ and later described in a paper.⁶ The graph in Fig. 3 is representative of our results.

The mouse consistently beat out the other devices for fast, accurate screen selection in our working context. For some months we left the other devices attached to the workstation so that a user could use the device of his choice, but when it became clear that everyone chose to use the mouse, we abandoned the other devices.

FIGURE 1
CDC3100 workstation
at which we
experimented with
selection devices.

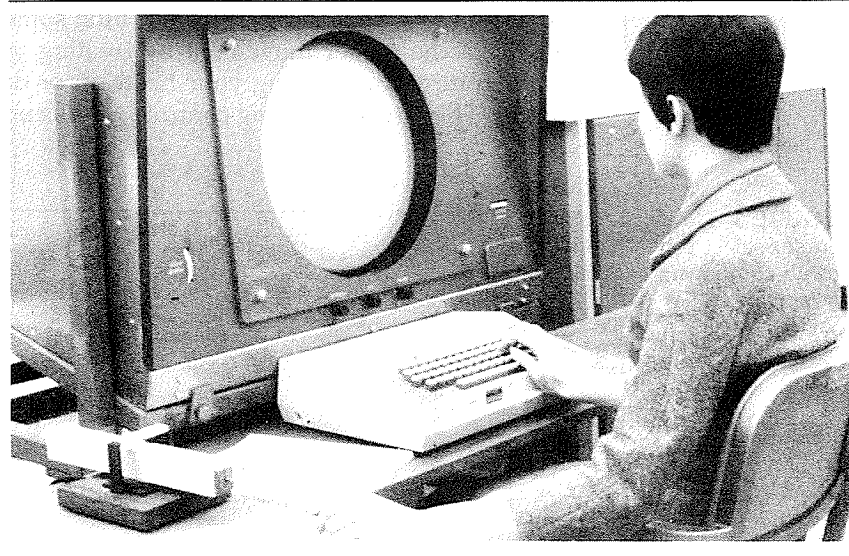
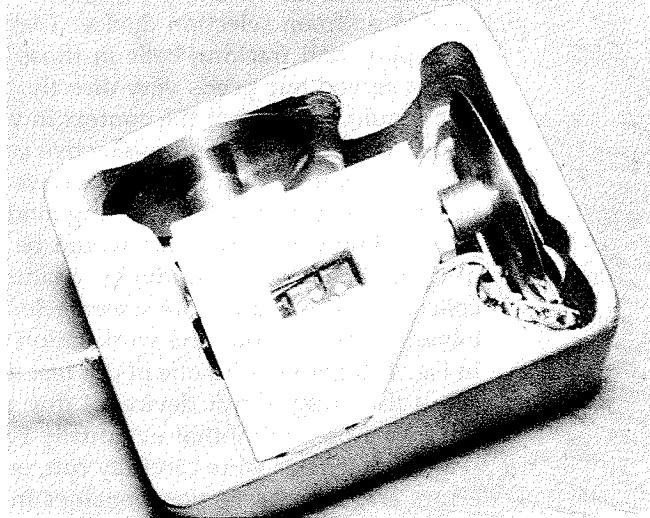


FIGURE 2
Bottom of wooden
"mouse," one of the
first selection
devices.



No one is quite sure why it got named a "mouse," or who first started using that name. None of us would have thought that the name would have stayed with it out into the world, but the thing that none of us would have believed either was how long it would take for it to find its way out there.

We also experimented with a way to move your knee up and down or swing it sideways to move the cursor, so you could keep your hands on the keyboard (Fig. 4). I also built something that allowed one to control the cursor movement by rotating your head for sideways cursor control, or nodding your head up or down for vertical cursor control. This "nose-pointing control" of the cursor left both hands free

FIGURE 3
Cursor-select test
graph comparing
various selection
devices.

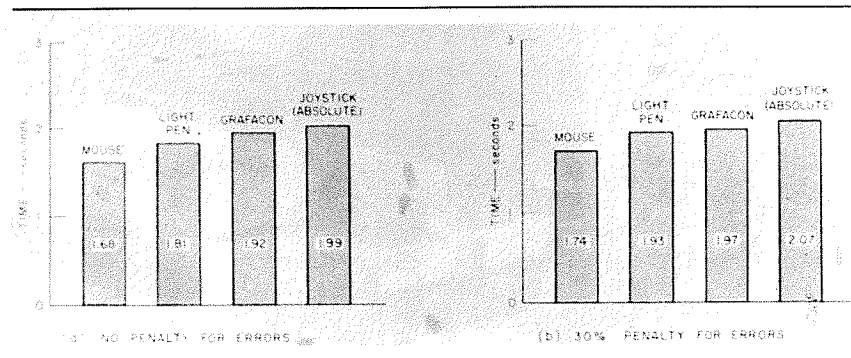
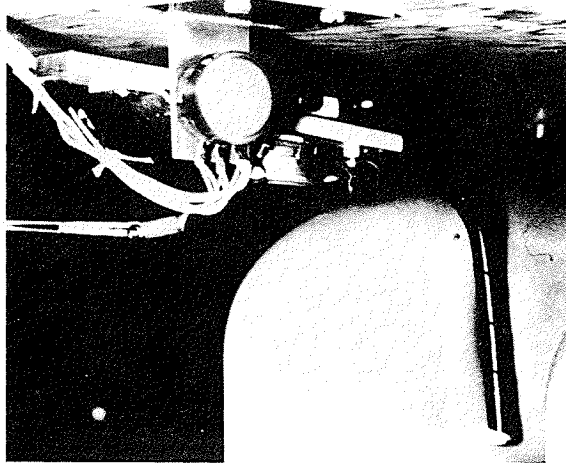


FIGURE 4
An experimental
knee control device.

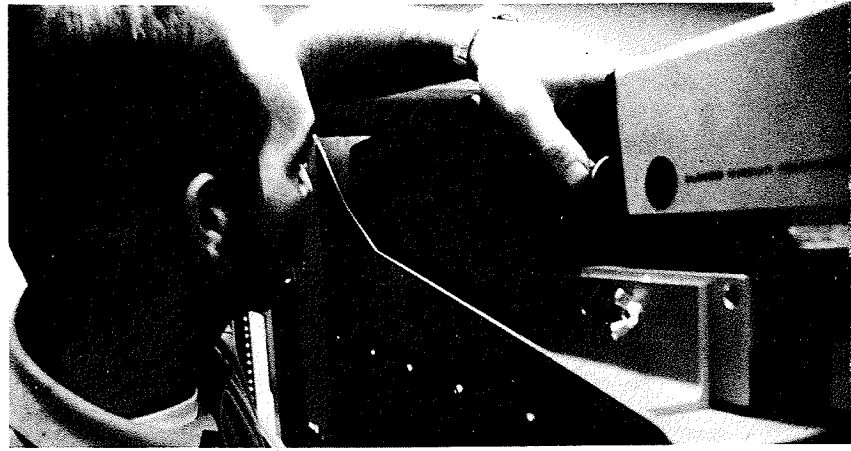


to operate the keyboard. In each case, some muscles would cramp up. Really, in almost any of these cases, you can get used to things like that. I'm sure there are a lot of things to explore in the future about what is going to work best for interfacing, so that you can tell the computer to which objects on the screen you want to direct its attention. However, we had many things to do, so it was a diminishing return for us to pursue these alternative devices—we stayed with the mouse. I have always assumed, though, that something better than the mouse is likely to emerge once the user market becomes more adventurous in exploring the means to higher performance.

OUR MULTI-USER COMPUTER AND DISPLAY SYSTEM, CIRCA 1968

The first time-sharing system we got was an SDS940. We considered a number of alternative ways to provide ourselves with a flexible, responsive display system, and finally designed our own. The computer and display drivers time-shared their attention among multiple, 5-inch CRTs, which happened to be the most economical size for a given percentage of screen resolution. In front of each CRT we added a commercial, high-quality video camera, mounted with a light shroud over the camera lens and CRT screen (Fig. 5). The resulting video signal, amplified and piped out to our laboratory, drove the video monitors that were our workstation displays. Two display generators, each driving up to eight CRTs, implemented with vacuum-tube technology were both bulky (Fig. 6) and very costly. It took one and a half people to keep those things running all of the time. The stroke-generated characters and vector graphics allowed us to have flexible, mixed text and

FIGURE 5
Our early computer
set up with 5-inch
CRTs.



graphic document presentations. The display generators were driven from a direct-memory-access channel that provided very fast (i.e., one refresh-cycle time) creation of a new display image. Figure 7 is a picture of one of our workstations—the TV monitor, with a little wooden table we made and a keyboard and a mouse and a keyset.

We explicitly designed for a detached keyboard. My workstation-design philosophy was to fix it so what you are looking at is positioned for best viewing, and the devices you use to control the computer should be located where it is best for you to operate them. Don't get caught in the anachronism that because we got used to paper and pencil and that technology, we ought to be able to have our controls right

FIGURE 6
The Augmentation
Project display
system.

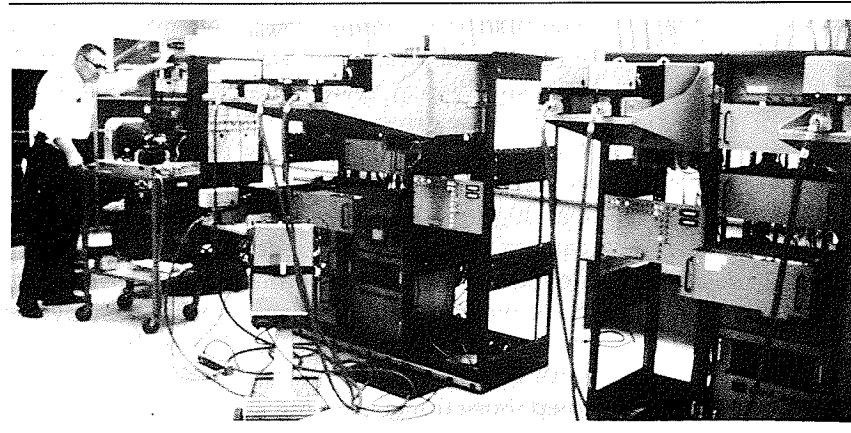
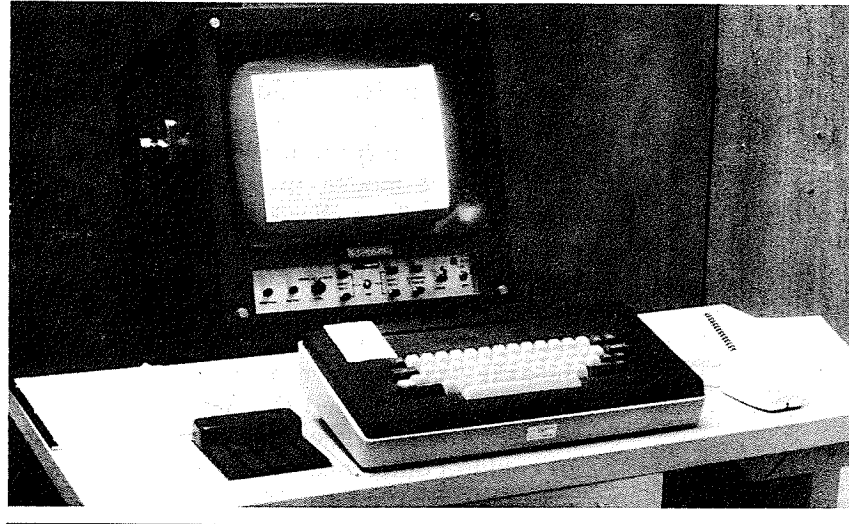


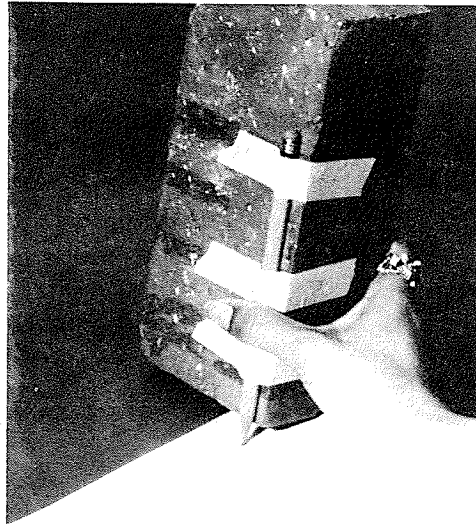
FIGURE 7
Basic workstation:
table with keyset,
detached keyboard,
mouse, and a
separate monitor.



on the surface of the thing that we're working on. It may end up that that way is best, but don't make an a priori assumption. So we didn't.

One way of explaining to somebody why it could make a significant difference if you can do things faster, is to provide a counter example. So, I had them write with a brick taped to their pencil (Fig. 8), because it's only a matter of happenstance that the scale of our body

FIGURE 8
Brick "writing"
device.



and our tools and such lets us write as fast as we can. What if it were slow and tedious to write? A person doesn't have to work that way very long before starting to realize that our academic work, our books—a great deal would change in our world if that's how hard it had been to write.

What if you speed it up? The keyset shown in Fig. 9, in combination with the mouse, provides a two-handed, higher-speed option. When you strike a chord, the computer interprets it as a character—not really distinguishing between characters generated by keyset or keyboard. Our chord-character code (a binary counting scheme mapped to the alphabet) is not really very difficult, my six-year old kids took less than a week to learn it. At SRI, we had a project in the early 1960s to experiment with computer-aided, psycho-motor skill training, and we used this keyset skill for the very first thing to be learned. The experiment blew up because everybody learned it so fast we couldn't differentiate between those who did and those who didn't have special computer aids.

Figure 10 is sort of the picture that I think about when I'm told that we've got to make it easy to learn. Tricycles are "easy to learn and natural to use"—no hard balancing problems. How much is it worth to you to extend your mobility? A few hours of learning to balance on two wheels?

By 1968 I had a workstation like the one in Fig. 11 in my office, and I just started "living" on it. In our lab we had six to ten more, time-shared among our group. The whole feeling was great. It was a very, very interesting sort of activity in those days. Who says we didn't do workstation research? This one, our "Yoga Workstation" (Fig. 12 with Bill Duval), got to be one of the favorites for some reason. This is what you saw all the time—people sitting there in their two-handed working mode, occasionally switching to type on a keyboard.

Figure 13 shows Bill English with a workstation made for us by the Herman Miller office furniture company, who were trying to experiment with us. This little console swivelled on one of the chairs and

FIGURE 9
Keyset input device.

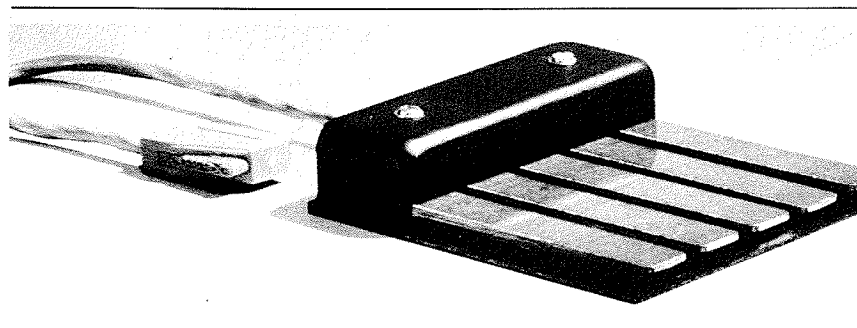
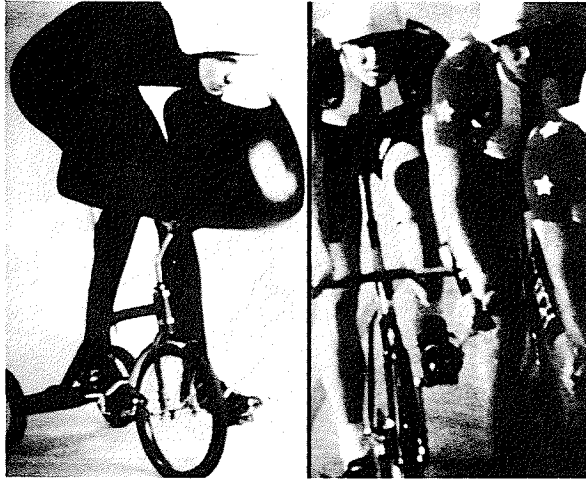


FIGURE 10
Trike/Bike.



went where you went; you could lean back and work with it very nicely. I wanted to show this workstation and Bill English, who deserves an immense amount of credit for getting things to work. I waved my hands and pointed, and somehow didn't make much progress. But Bill really made things work in those days, so I owe him a lot.

FIGURE 11
Doug's office
workstation
replicated 6-10 times
in the lab.

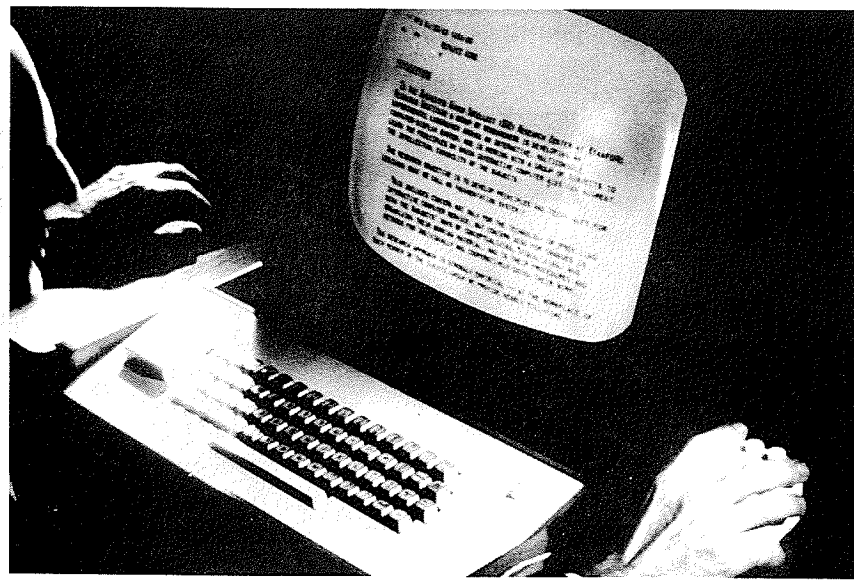
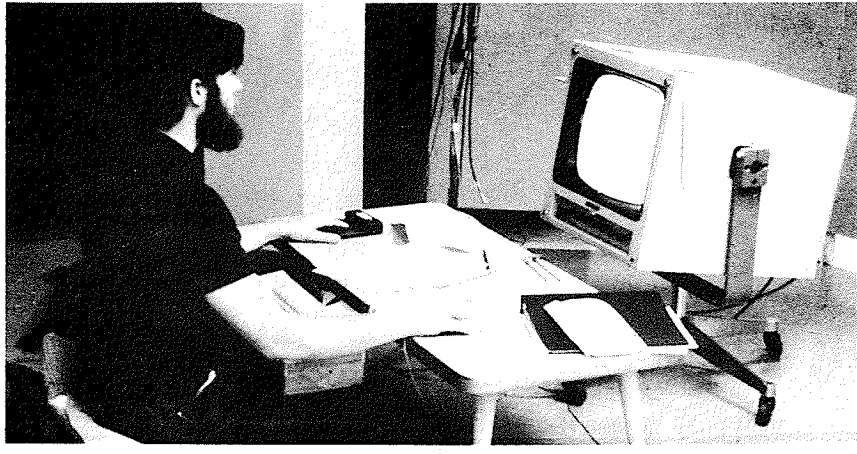


FIGURE 12
Yoga workstation—
Bill Duval is pictured
here.

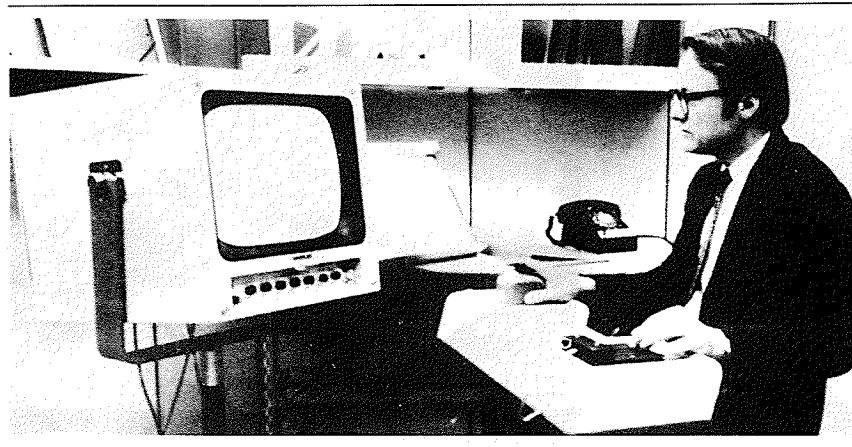


OUR "BIG SHOW"—THE 1968 FALL JOINT COMPUTER CONFERENCE

By 1968 we had a marvelous system. We called it "NLS" (later "AUGMENT"). "NLS" stood for "online system," in contrast to "offline system," which we dubbed "FLS" just to have different acronyms. A few people would come and visit us, but we didn't seem to be getting the type of general interest that I expected. I was looking for a better way to show people, so we took an immense risk and applied for a special session at the ACM/IEEE-CS Fall Joint Computer Conference in San Francisco in December 1968.

We set up to give an online presentation using a video projector pointing at a 20-foot screen. Brooks Hall is a large auditorium, and that

FIGURE 13
Bill English at our
Herman Miller
designed
workstation.

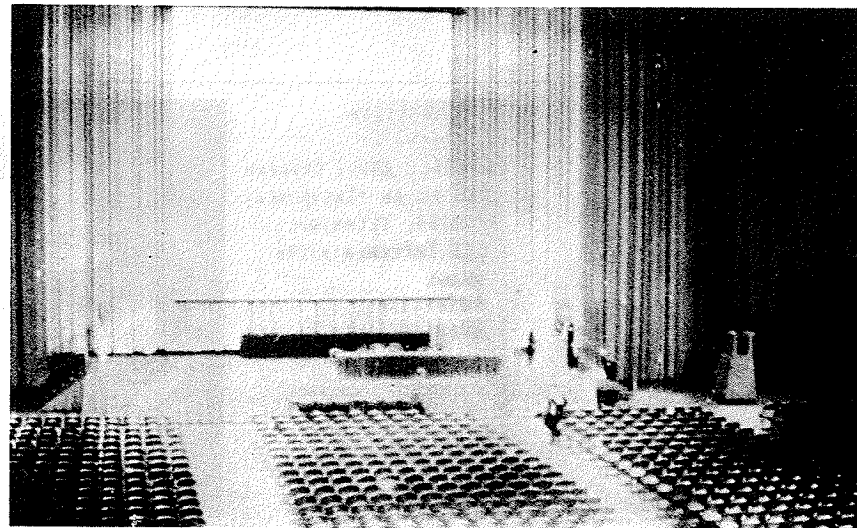


video projector could put up our display images so you could read them easily from up in the balcony (Fig. 14). The video projector we rented (built by a Swiss company, Eidophor) used a high-intensity projection lamp whose light was modulated by a thin film of oil, which in turn was modulated by the video signal. On the right side of the stage, I sat at our Herman Miller console. We set up a folding screen as a back drop behind me. I saw the same image on my workstation screen there as was projected for the audience to see.

We built special electronics that picked up the control inputs from my mouse, keyset, and keyboard and piped them down to SRI over a telephone hookup. We leased two microwave lines up from our laboratory in SRI, roughly 30 miles. It took two additional antennas on the roof at SRI, four more on a truck up on Skyline Boulevard, and two on the roof of the conference center. It cost money—running that video projector, and getting people to help us do all of that, cost money; making the special I/O cost money; and leveraging special remote-presentation technology on top of our advanced, developmental laboratory technology created extra risk—and I was using research money. The nice people at ARPA and NASA, who were funding us, effectively had to say, “Don’t tell me!” because if this had flopped, we would have gotten in trouble. But anyway, it worked, and the main reason was because of Bill English’s genius for making things work.

Back in our lab, we dismantled a number of the display units in our display system, so that we could use the cameras in San Francisco and SRI. We borrowed a few tripods and got some extra people to be

FIGURE 14
Auditorium of the
ACM/IEEE-CS Fall
Joint Computer
Conference,
December 1968.



camera people. One of our friends, Stuart Brand, who was at that time working on his first *Whole Earth Catalog*, helped as well. So it was really a group project; there were about 17 of us.

On my console on the stage, there was a camera mounted that caught my face. Another camera, mounted overhead, looked down on the workstation controls. In the back of the room, Bill English controlled use of these two video signals as well as the two video signals coming up from SRI that could bring either camera or computer video. Bill could select any of these four video images with optional mixing and frame splitting. We had an intercom that allowed him to direct the action of the people in our lab at SRI who were generating computer images or handling the cameras sending the video up from SRI.

We didn't use any specially made system capabilities; we were just using NLS the way it worked at that time. It had mixed text and graphics, so we could use those to display and represent things. We had the agenda in NLS, and we could run different parts and show diagrams; we could do things as examples. So it was a mix of things of: here's the script and stuff to tell you about, and here's the way it runs; we could also bring other display screens or faces, from our SRI lab, in and out on the screen. At that time we firmly imagined that this was the way future conferences would be run.

We could do screen splitting. Figure 15 shows the agenda list with a little marker to show that I'm between two particular items. In Fig. 16, I make a temporary (shopping) list. This was the beginning of our demonstrating ways of structuring ideas. The NLS system supported the user in getting the list organized into categories.

We wanted to show how the mouse works. The projected video showed Don Andrews controlling a cursor from our SRI laboratory by

FIGURE 15
Still frame from the FJCC '68. Split screen with Doug on the right and demo agenda on the left.

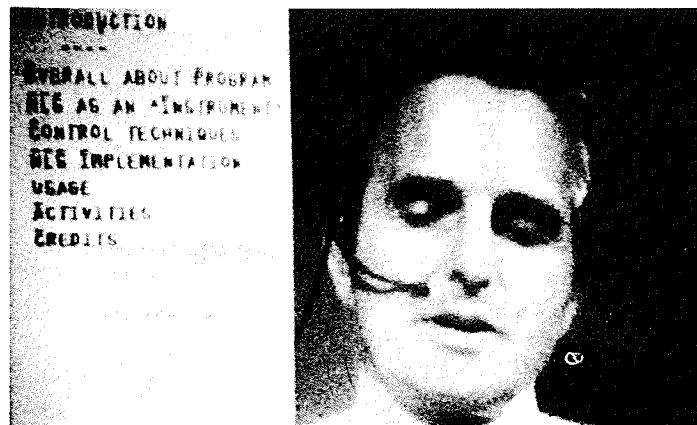
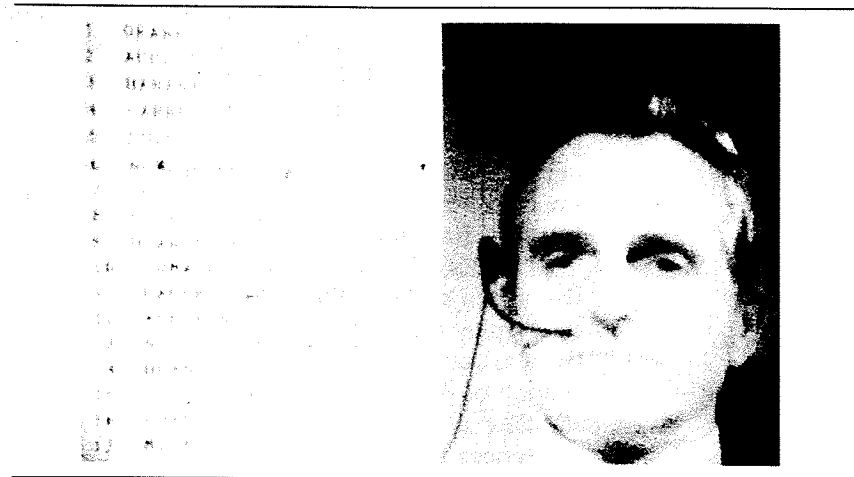


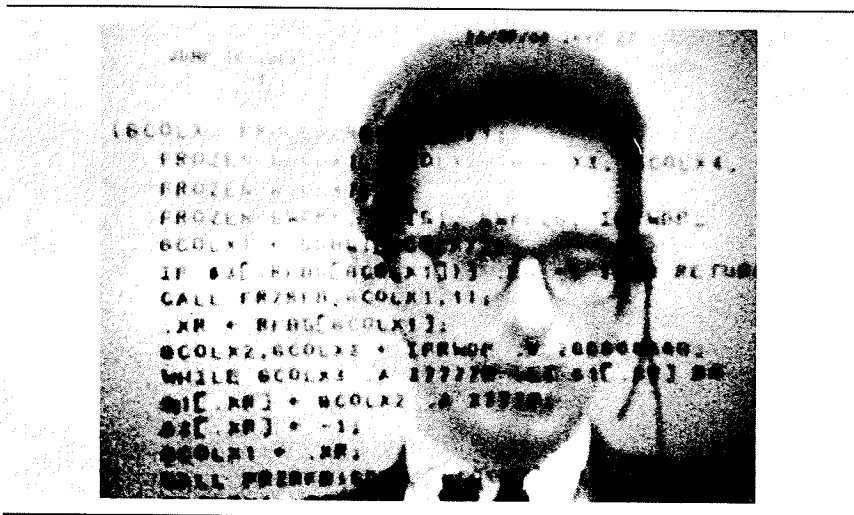
FIGURE 16
The text is now an unstructured shopping list.



moving his mouse around. The superimposed video image of the display screen showed that the cursor would follow it exactly to show how the wheels worked. Remember, this was 1968—the first public appearance of a mouse. I could also show how the simultaneous use of mouse and keyset worked in such a way that the audience could watch my hands in the lower window and see the computer action in the upper window.

Then we brought in Jeff Rulifson to tell about how the software works (Fig. 17). At the same time, his face could be brought in and out

FIGURE 17
Jeff Rulifson can be seen on the screen mixed in behind code of a procedure.



behind the display image that he was working with, demonstrating NLS's power for working with very explicitly structured software. He showed graphical diagrams that were embedded in the source-code documentation. During Jeff's presentation, Bill English brought the picture from a laboratory camera that caught the view (Fig. 18) of Jeff's keyset operation as he was manipulating his demonstration images—unconscious and unhurried—a nice way to show the fluid speed offered by combined mouse and keyset use.

Toward the end, we also showed that we could cut a "hole" in the screen and see Bill Paxton's face from SRI (Fig. 19). For the computer-display part of the screen, we could switch back and forth between his work and mine; and we could also switch which of us was controlling all of this.

The associated FJCC publication about NLS⁷ and other relevant references are listed at the end of this paper.

[*Editor's Note:* Engelbart's colleagues—Bill English, Charles Irby, Jeff Rulifson, Bill Duval, and Bill Paxton—all found themselves at Xerox PARC in the 1970s, creating a personal workstation that embodied these ideas.]

NLS ENHANCEMENTS—INTO THE MID-1970s

You can't imagine the relief when it worked. It went on for 90 minutes, and afterward we thought for sure that the world would be talking about everybody starting to augment now. Well, it didn't happen, but we went ahead anyway. I want to discuss a few of the things we did

FIGURE 18
Jeff's keyset and
hand behind some
lines of text.

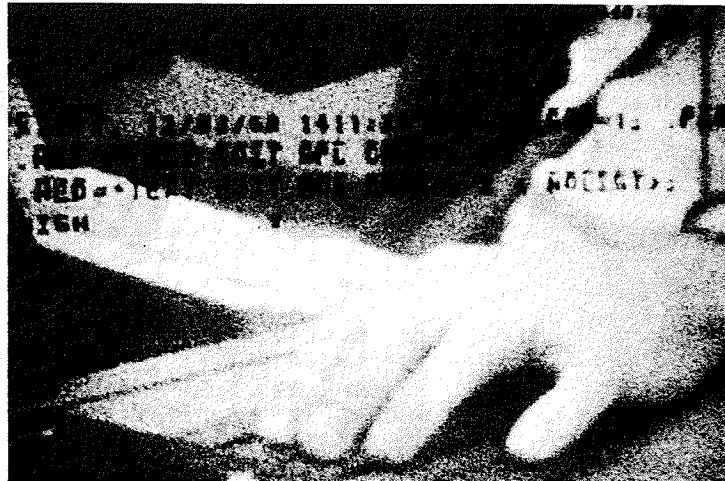
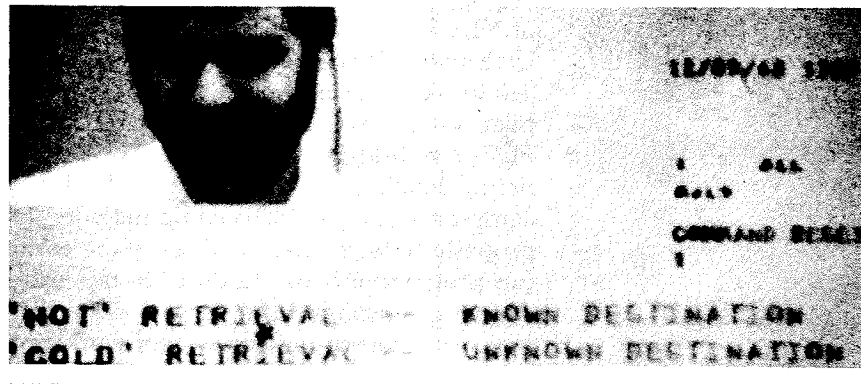


FIGURE 19
View of text retrieval
with image of Bill
Paxton in
background.



physically to the system after that, and then go into some of the conceptual framework.

1969

- We began design of windowing capability for NLS.
- We developed concept of a user "reaching through" his personal work place (i.e., his familiar online working files and application programs) to access less basic, specialized data and application processes (and other people); that is, the "reach through" should provide access to these, translated by the integrated support system, so as to appear as coherent parts of his familiar, personal work place.
- We specified our first mail and "Journal" system as part of an explicit pursuit of a "Dialog Support System," planning for it to be part of our ARPANET-NIC service.
- We developed document-outputting capability processing our composite, text-graphic document files to drive a service-bureau, CRT-based, full-page, Stromberg-Carlson photo printer to produce documentation with graphics and text mixed on the same pages.
- We became the second host on the ARPANET with our SDS 940. (UCLA was first, UCSB next, then the University of Utah, then. . . .)

1970^s

- Detailed use of NLS began for internal management processes of ARC: cost records, working forecasts, purchase requisitions, and so on.

- We began using the ARPANET to facilitate our reprogramming of NLS for the forthcoming PDP-10 TENEX. The University of Utah had a TENEX on the network, and we used NLS on the 940 to write our new PDP-10 code; using our tree-meta compiler, we developed a cross-compiler for our 940 that produced PDP-10 relocatable binary code. We would ship that over the net for loading and debugging on Utah's TENEX. When the two computers and the intervening network link were all working properly (lots of "flat tires" as in the early days of automobiles), our programmers would do all of this back and forth transitioning "through" the same workstation. I think that it was not only a record-making way of working, but the NLS transport task was accomplished in remarkably short time (we attributed part of the efficiency to the network, and part to the use of NLS).
- We brought NLS up on the PDP-10 TENEX with improved and new features (including multiple windows). The transfer process, and a detailed description of the design changes and new features for NLS are described in a June 1971 technical report.⁹
- We began using our Mail/Journal system within our group. Integrated into NLS, this assumed that a mail item was a document—so any part of all of an NLS document could be sent—and provided for a permanent record in explicitly retrievable form (our Journal). As an electronic-mail system, this was quite advanced. It had a directory service (our Ident System) to provide mail-relevant information about registered users; mail distribution was addressed by people's Idents, with no need to know or specify which host they used. Fields were provided for superceding other items, and for attaching keywords. An on-line index was provided for stored items.¹⁰

1972

- We began developing our first, integrated Help system.
- We formulated the "AKW Architecture," implemented in stages.¹¹
- We implemented the "shared-screen," televising mode of on-line collaboration between two or more NLS users.¹²

1973

- We brought up a table subsystem in NLS.
- We designed our first, totally modular user interface system, as later described in the *OAC '82 Digest*,¹³ and got it running on

a PDP-11 that talked to our TENEX through the network, via our procedure call protocol.

- We developed our line processor, as described by Don Andrews.¹⁴ It incorporated Intel's first microprocessor (the 4004) in a special box that was inserted in the communication line between a dumb display terminal and a modem. This made use of our virtual terminal protocols, and managed a multi-window, two-dimensional screen using off-the-shelf, "dumb" display terminals. Our mouse and keyset input devices were plugged into the line processor, which appropriately translated their actions to control cursor position and special communications to the host. A printer port on the line-processor provided local printout service; a special communication protocol allowed the host to send printer packets mixed in with display-support packets.
- We finalized specification for our network virtual terminal, something that has become a key part of our architecture. The objective, on the one hand, was to free the application programmers from worrying about the special features of different workstations, and on the other hand, to enable more flexible evolution by users of workstations they may adopt to fit particular needs. As part of this, there was a terminal-independent display manipulation protocol for communication from application program to terminal, and an application independent input protocol for communicating from terminal to application program.
- We generalized the file structure of our document files to provide for generalized property structures associated with each addressable object, intended to accommodate composite integration of such as graphics, digitized speech, scan-coded images, or any other arbitrary data form.

1974

- We gave up our high-performance, local display system for the line-processor supported, remote display system—to make ourselves live with the same remote services as our NIC clients and Utility customers. (On principle, we gave up our integrated, direct-view graphics and the fast response of our direct-memory-access, local display generator.)
- We opened our "Workshop Utility Service." Delivering NLS service over the ARPANET to DOD customers as pilot applications of office information service. We had gone out on bid for

commercial time-sharing services, selected Tymshare Inc. of Cupertino, Calif.; their host, named Office-1, provided the computer service. We fielded special trainers and application development staffs and cultivated special customer representatives into a spirited community.

1975

- We implemented our new, integrated graphics system, which could support remote display and manipulation of illustrative graphics on a Tektronix 4014 storage-tube display plugged into the line-processor's printer port. Figure 20 shows the graphic-station setup used for development in our lab. Bob Belleville, to the right in the picture, developed this "new" graphics software. (He subsequently went to Xerox and helped them with the Star hardware, and then was the project manager for the Apple MacIntosh.)

In 1973 we had generalized our file structure with provision for attribute-value property structures associated with each hierarchic node. We could then embed arbitrary data objects in our documents. The first utilization was to reinstate embedded graphics, extending our hierarchical list structure and its pointers to the text objects with, for example, an associated pointer on one of the text objects to a graphic substructure for the associated illustration. The user's concept would be shown in an associated "document-page" image. We assumed that soon we would be using digitized speech strings and executable object code as part of our composite documents. And some day, when storage would be cheaper, we would even be embedding scanned images.

An illustration produced as a plotter-driver file by any other graphics system could be picked up and attached to a specified location in an NLS document, and could be subsequently viewed and modified

FIGURE 20
Bob Belleville at Tek
graphic workstation.

