Before the Altair: The History of Personal Computing

Article *in* Communications of the ACM · September 1993

DOI: 10.1145/162685.162697 · Source: DBLP

1 author:

Larry Press
California State University, Dominguez Hills
99 PUBLICATIONS **963** CITATIONS

SEE PROFILE

Before the Altair: The History of Personal Computing

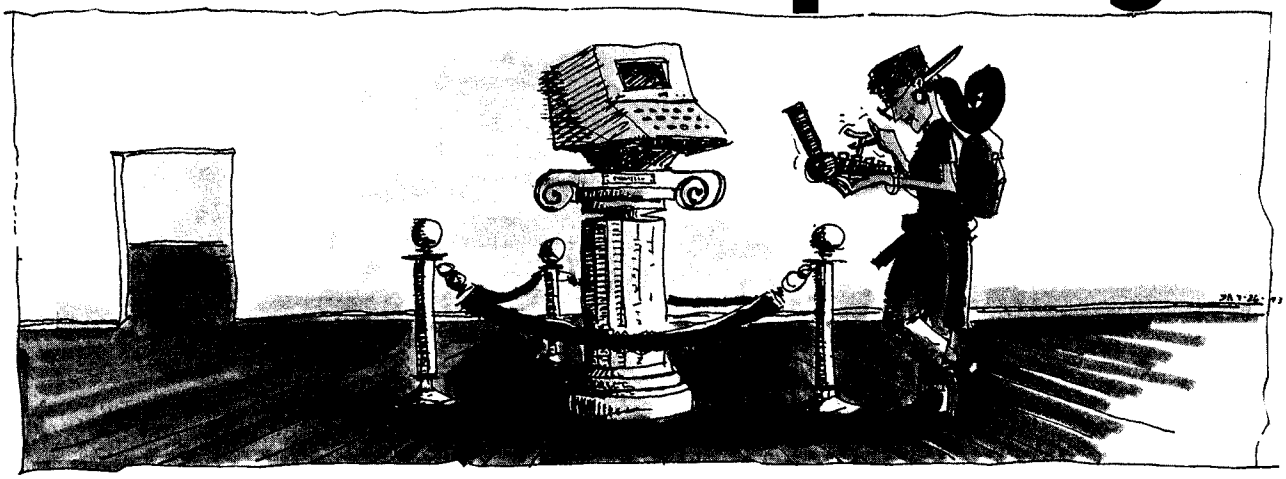

ILLUSTRATION: DENNIS AHLRICH

Imagine a modern personal computer. It has a direct-manipulation, WYSIWYG user interface, and programs for drawing, painting, writing, and other tasks. It is connected to a LAN with email, database, print, and other services. The LAN may be connected to other LANs, forming an internet, and it may be connected to a wide-area network like the Internet.

That also describes the Xerox Palo Alto Research Center (PARC) nearly 20 years ago [24]. Today's personal computers are designed and sold by Apple, IBM, and others, but researchers had developed the ideas and techniques they use by the mid-1970s. This column discusses nine of those early ideas, the people who had them, and the labs where they worked.[1] We begin with the observation that an information processing machine was needed to augment intellect.

Augmenting Human Intellect

In an *Atlantic Monthly* article published just after World War II, Vannevar Bush, a prominent researcher at MIT, stated a problem: scientists were increasingly unable to keep up with professional information. They could not find or read everything in their fields, and what they did read (and write) they often forgot or misfiled because of inflexible indexing systems. Having stated a problem, Bush postulated a machine to solve it. He called it "memex" because it was to extend the powers of human memory and association:

A memex is a device in which an individual stores all his[2] books, records, and communications, and which is mechanized so it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory [3].

He envisioned the memex as a desklike machine with a keyboard, knobs, levers, and two displays. Storage would be on microfilm, and dry photography would be used for input. (Bush was working on microfilm-based library information retrieval when the article was written.) Most important, documents were not just indexed and stored. They could be annotated, and unlimited linking of documents was to be possible.

Bush wrote the first draft of this article before the war, and was concerned that it be published where it would be read by influential people. His most important reader turned out to be Doug Engelbart, a young sailor who "avidly read it in a Red Cross library on the edge of the jungle on Leyte ... in the Fall of 1945" [18]. Engelbart saw the same problem as Bush, but his solution was not

Larry Press

[1]The ideas and projects I have included are not necessarily the earliest. For example, the Dartmouth remote computing demonstration, or the packet-switched LAN at the National Physical Laboratory in Middlesex, England, are not mentioned because they did not have the scope and influence of the ARPANET or Ethernet. I am counting on readers to fill me in on oversights.

[2]Masculine pronouns were commonly used at this time. I must confess that I once thought dropping that usage was a silly gesture, but it is now jarring to see it, and clearly unfair.

microfilm, it was "having the computer do some of his symbol-manipulating processes for [man] so that he can use more powerful concepts and concept-manipulation techniques" [4]. Engelbart elaborated a theoretical framework of knowledge work, knowledge representation, and of the relationship between people, societies and tools [4]. He saw knowledge work as building structured concepts—networks of text frames and drawings—which might represent the work of a few hours or of a few years. Similar structures could represent procedures, similar to computer programs or project plans. He emphasized that people often built these structured concepts in groups, and pointed out that tools, people, and societies coevolve. We shape our tools and they shape us.

The report also speculated on a machine, a two-display workstation with a light pen and two single-handed keysets, and specific applications. This framework guided 15 years of research and prototype development in Engelbart's lab at the Stanford Research Institute (SRI). Bootstrapping development by using their own tools, Engelbart's group began the field of computer-supported cooperative work, and invented WYSIWYG word processing, the mouse, multiwindow displays, electronic meeting rooms, and more. The system they developed is described in [5, 6], and I wish I had been able to use it to write this article.

Interactive Computing

As Table 1 shows, the way we use and communicate with computers has changed over time. Early computers were expensive, so it was important that they not sit idle. Programs and data were prepared off-line, and fed to the computer in batches by operators. After a job ran, the output returned to the user. This approach kept the computer busy, but wasted the user's time. Every programmer who got a hands-on "shot" at the console for debugging or typed a five-minute "hello world" program into an early timesharing system knew it was better to work interactively than in batch mode.

Interactive computing bloomed at MIT in the 1950s, beginning with Jay Forrester's Whirlwind computer. Whirlwind processed real-time telemetry data and was used interactively by an operator at a display-based console. It was followed by other experimental interactive computers at MIT. All of the MIT engineers using them understood the value of interaction, but J. C. R. Licklider spread the vision. In an influential paper on man-machine symbiosis, he wrote:

Computing machines can do readily, well, and rapidly many things that are difficult or impossible for computers. That suggests that a symbiotic cooperation, if successful in integrating the positive characteristics of men and computers, would be of great value [13].

This paper was widely read and laid the groundwork for the study of what we now call "human-computer interaction." (Licklider was a research psychologist.) For example, the paper outlines scientific visualization, voice I/O, and pen-based interfaces.

While people at MIT enjoyed interactive computing, the rest of the world was submitting jobs and waiting for output. Timesharing brought interactive computing to the masses. The SAGE air defense system, which evolved out of the Whirlwind, timeshared a computer between many users working at the same task, but the first description of general purpose timesharing was in a widely quoted paper by Christopher Strachey, who wrote:

... several operators are using the machine during the same time. To each of these operators the machine appears to behave as a separate machine (smaller of course, and slower than the real machine) [22].

Strachey describes a system to do this, and outlines architectural extensions that would be needed for timesharing [22]. These included prioritized interrupts, memory protection, and supervisor-only instructions. Strachey's paper was published in 1959, and by the mid-1960s there were at least six experimental timesharing systems.

Much of Engelbart's work and several of the timesharing experiments were funded by the Advanced Research Projects Agency of the Department of Defense (ARPA). This was no coincidence—Licklider was the first head of the Information Processing Techniques Office at ARPA.

Direct Manipulation

Timesharing enabled interactive computing, but as long as we worked at printing terminals, we were restricted to command-oriented user interfaces. The immediacy of direct manipulation, WYSIWYG interfaces required a display and pointing device. Like interactive computing, the power of direct manipulation was obvious; it only had to be experienced to be appreciated.

This was understood early at MIT. For example, Doug Ross tells of a program he wrote in 1954 [7], which let him draw on the display screen using his finger. However, it remained for Ivan Sutherland's Sketchpad program to demonstrate the power of direct manipulation to the world. Sketchpad, written for the TX-2, a descendent of Whirlwind, was a CAD program in which the user drew directly on the screen using a light pen. Sutherland described Sketchpad and its data structures, algorithms, and applications in [23], where he wrote:

A Sketchpad drawing is entirely different from the trail of carbon left on a piece of paper. Information about how the drawing is tied together is stored in the computer as well as the information which gives the drawing its particular appearance.

Sketchpad was arguably the most innovative program ever written. It opened the field of computer graphics, and featured software inventions like the cursor, the window (zooming, not scrolling), clipping, rubberband drawing, constraint-based drawing, and gesture recognition. Furthermore, it was object-oriented. Anything you drew could be used as a "master" (i.e., class definition), from which instances (his term) could be derived. Instances inherited the properties of the master, and could be modified, resized, dragged, rotated, and so forth. They could

Computers evolved by merging programmable devices and calculating machines. The gap was closed by the ENIAC in 1946. Nearly 50 years later, we see the merging of two other streams of development, computing and communication.

themselves be masters, and they could be recursively combined to form compound objects.

While Sutherland worked on drawings, Engelbart applied direct manipulation to text. In [4] he proposed (and later implemented) a hypothetical word processing program with automatic word wrap, search and replace, user-definable macros, and commands (gestures) to move, copy or delete single words or arbitrarily selected blocks of text. When the text filled the screen, it scrolled. When changes were made "the text simultaneously readjusted to present the neat, no-gap appearance it had had" [4]. Margins could be adjusted at will, and text could be laid out in dynamically resizable columns. There was also to be an online dictionary with definitions, synonyms and antonyms.[3]

Computers for Everyone

Early computers were expensive and complex, so it took special vision to see they would one day be widely used. John Kemeny and Thomas Kurtz of Dartmouth College had that vision. During 1962, they and their students implemented several simple programming languages on an LGP-30 computer, and by 1963 they had made the decision that all Dartmouth students would be taught computing. They later wrote, "The primary goal motivating our development of DTSS was the conviction that knowledge about computers and computing must become an essential part of a liberal education" [10].

Following a design policy that "In all cases where there is a choice between simplicity and efficiency, simplicity is chosen" [11], they built the DTSS, the Dartmouth timesharing system and invented Basic. Basic was designed for interactive use by beginners. It had interactive I/O statements, and built-in string functions made it easy to parse input and build conversational programs. An integrated monitor allowed easy interactive debugging.

While Kemeny and Kurtz were bringing computers to university students, Bob Albrecht, brought them to "the people." He began teaching Fortran to high school students in 1962, and "We had so much fun we extended the program to eight other schools" [1]. Having smelled the flowers, Albrecht soon left his job at Control Data and moved to California, where he established the People's Computer Center (PCC). PCC was a walk-in center where children and adults played games, wrote Basic programs, and demanded that computers be used "by the people, not against them." Many future Silicon Valley entrepreneurs were regulars at PCC's weekly potluck dinners. Albrecht was also influential through his writing. He published a tabloid periodical, and his uniquely informal books were widely read. (*My Computer Likes Me When I Speak BASIC* sold over 250,000 copies.)

In [19], Seymour Papert described his experience using Logo to teach junior high students mathematical and procedural thinking. Alan Kay was influenced by Papert, and when he began work on personal computing at PARC, he wrote, "We wanted our range of users to include children from age 5 or 6 and 'noncomputer adults' such as secretaries, librarians, architects, musicians, housewives, doctors, and so on [9]. (Kay's group at PARC was called the Learning Research Group.)

Computers are Communication Devices

Computers evolved by merging programmable devices and calculating machines. The gap was closed by the ENIAC in 1946 (see Table 2). Nearly 50 years later, we see the merging of two other streams of development, computing and communication (Table 3). Today many of us believe computers are primarily communication devices, but Licklider understood this in the early-1960s. He and Robert Taylor wrote "the use of the computer as a communication device . . . promises to bring a new depth of intellectual interchange to the fine old art of fact-to-face communication" [14].

The user communities which had formed around the email, bulletin boards, and subroutine libraries of the early timesharing systems hinted at the role of computers in communication. Licklider and Taylor foresaw the possibility of extending those communities:

What will on-line interactive communities be like? In most fields they will consist of geographically separated members, sometimes grouped in small clusters and sometimes working individually. They will be communities not of common location, but of common interest. *As attractive as this vision was, they also asked: Will 'to be on line' be a privilege or a right? If only a favored segment of the population gets a chance to enjoy the advantage of 'intelligence amplification,' the network may exaggerate the discontinuity in the spectrum of intellectual opportunity [14].*

Inspired by telegraph systems using torn paper tape, Paul Baran of RAND wrote a series of memos proposing a packet-switched military network [2]. His memos present a packet format, routing algorithm and switching computer and microwave link design. They also discuss robustness when nodes or links fail or are destroyed, encryption, staffing difficulty, and other topics. While the network would be complex and diffi-

[3] I do not know which of these features were invented by Engelbart and which existed in other early programs, such as Expensive Typewriter at MIT.

Table 1. Evolving computing/communication paradigms

Computing Paradigm	Communication Paradigm
1950s: batch processing	transmit batches of jobs
1960s: timesharing	interactive terminals
1980s: desktop computers	local-area networks
1990s: portable computers	wide-area networks

Table 2. Inventions combining calculation and programming

Project	Technology
1838 Analytical Engine	mechanical
1890 Unit-record Machines	electromechanical
1946 ENIAC	electronic

Table 3. Inventions bridging computing and communication

Project	Technology
1950 Whirlwind/SAGE	electronic
1960 Timesharing	electronic
1969 ARPANET	electronic
1973 Alto/Ethernet	electronic
1996* The Net	optronic/wireless

**I picked 1996 because the technology and infrastructure are falling into place now, and 1996 is 50 years after ENIAC. The technical and conceptual evolution between the Gutenberg Bible and the relatively inexpensive, portable, "pocket" books of Aldus Manutius also took around 50 years.*

Table 4. Common data types. Research with these data types began in the 1950s, but most required cheaper devices for commercial application.

Type	Commercial Application	Example
Numeric	1950s	scientific and engineering data
Alphanumeric	1950s	DP/MIS files
Control signals	1960s	process control signals
Text	1970s	word processing and email documents
Knowledge	1980s	expert system rules
Image	1980s	graphs, paint, and draw images
Motion Video	1990s	animation and recorded video
Music	1980s	MIDI scores
Voice	1990s	recognition and synthesis
Ink	1990s	handwritten characters

cult to build, he predicted it would solve many communication problems, and save money. Baran's vision was less sweeping than Licklider's, but his analysis was accurate.

Licklider and others encouraged ARPA to fund the deployment of a national network, the ARPANET. Four nodes began operation in 1969, and the rest is history. Today, the Net has thousands of mailing lists, listservers, news groups, electronic journals, and other "communities of common interest."

When researchers at PARC decided to build an experimental personal computer, the Alto, they wanted to duplicate the sense of community surrounding timesharing systems. This led to a LAN, the Ethernet being an integral part of the Alto design. Around 1,000 networked Altos were eventually deployed at PARC, and this experience led to the invention of client-server computing and the bit-mapped laser printer. The Alto designers wrote:

The high bandwidth communication provided by the Ethernet has been more valuable than anticipated, since we underestimated the importance of servers. The network and network services have been the mainstays of the environment, and we feel that a facility with an order of magnitude lower bandwidth would have had a qualitatively different effect [17].

Many Data Types

One way to view the evolution of digital information processing is to note when various data types became commonplace (see Table 4), but of course research precedes deployment. We have already discussed Sutherland and Engelbart's research with graphics and text. As early as Leibnitz, researchers (including Bush) had sought notation that would permit automatic deduction, and computer-based knowledge representation began in the 1950s. For example, John McCarthy wrote "If one wants a machine to be able to discover an abstraction, it seems most likely that the machine must be able to represent this abstraction in some relatively simple way"[15]. He then invented Lisp. Research on voice recognition and synthesis, music synthesis, computer art, and computer ani-

When researchers at PARC decided to build an experimental personal computer, the Alto, they wanted to duplicate the sense of community surrounding timesharing systems. This led to the LAN, the Ethernet being the most integral part of the Alto design.

mation was conducted at Bell Labs between the late-1930s and mid-1960s. The Illiac computer was used in music composition in the 1950s, and experiments in pen-based handwriting, symbol and gesture recognition were carried out at the System Development Corporation and RAND in the early-1960s.

While these and other early experiments were important, Kay's vision of a Dynabook dramatically integrated data types:

Suppose [the Dynabook] had enough power to outrace your senses of sight and hearing, enough capacity to store for later retrieval thousands of page-equivalents of reference material, poems, letters, recipes, records, drawings, animations, musical scores, wave-forms, dynamic simulations and anything else you would like to remember and change [12].

One or More per User

Since at first people shared computers, the idea that everyone should have their own was a breakthrough. Kay's Flex Machine [8] was intended to be a personal computer, as were the Altos at PARC. Designed for a single user, the Alto had a page-size (875-line, 70 dpi), bit-mapped display, a mouse, keyboard, five-finger keyset, Ethernet controller, and a removable disk pack. It was optimized for I/O—the user program and device controllers timeshared the system microcontroller and memory, and the firmware supported BitBlt, windowing, and a programmable cursor.

Alto software included several development systems and programs for drawing, painting, animation, music, email, word processing, and page layout, all with modern graphical interfaces. (Strategic employees at Apple and Microsoft came from PARC.) As in Engelbart's lab, the Altos, LAN, servers, and software were in daily use by the people developing them. By 1979, there were roughly 1,000 Altos, several dozen print servers, and 25 Ethernets connected by 20 Alto-based routers. In evaluating the project, the designers wrote:

The Alto has led to an entirely new kind of computing environment, because it puts computing power near the user, and makes it possible for him to do most of his work without relying on a centralized facility. ... One of the Alto's most attractive features is that it does not run faster at night [24].

Portability

Alan Kay has remarked that had Vannevar Bush enjoyed working in all-night coffee shops, he would have invented the portable computer. But Bush worked at a desk, and the vision of a portable computer was Kay's. For him the Alto was a vehicle for prototyping the real "Holy Grail," a powerful portable computer. Desktop computers were chained to desks like Gutenberg's first books, and Kay wanted a portable computer his mother could take to the market with her shopping list:

Several years ago, we crystallized our dreams into a design idea for a personal dynamic medium the size of a notebook (the Dynabook*) which can be owned by everyone and has the power to handle virtually all of its owner's information-related needs. ... We envision a device as small and portable as possible which could both take in and give out information in quantities approaching that of human sensory systems [12].*

Kay was also inspired by the shrinking calculators of the time. During the 1960s, Wang, Tektronix, and HP developed electronic calculators for the desktop, and in 1971 HP introduced a hand-held scientific calculator. Computerized shopping lists suddenly seemed a bit more likely.

A Mass Market

There were several early, commercial personal computers, most notably the LINC, the IBM 5100, and the Tektronix 405X, but the first practical, mass-market machine was the MITS Altair, sold as a $397 kit [21]. To be useful an Altair or its clones required a terminal, floppy disks, and a printer, which brought the price into the $5,000 range, but for that price you had a computer that did real work like word processing, file management, and running Basic, Fortran, Cobol, and PL/I programs. Professionals could own their own tools. Entrepreneurs flocked to the Altair, and personal computing was on its way.

The personal computer industry had a foundation to build on. The first years were like a fast-forward replay of the computer and word processing industries—assemblers, high-level languages, operating systems, superscalar CPUs, and modern applications, were implemented as soon as falling cost allowed. The similarity of today's systems to the Alto, is an indication that the computer industry has consumed much of the intellectual capital developed at MIT, SRI, and PARC. I wonder which labs are inventing the next 50 years?

Acknowledgments

Thanks to Alex McKenzie and Bob Anderson for their help with this column.

Pointers

- ACM held a conference on the History of Personal Workstations in 1986. Many of the people I have mentioned were there, and all the presentations and informal discus-

sions are found in [7], a terrific book.

• ACM held a conference on the History of Programming Languages in 1978 [25]. Two of the papers were on Basic and Joss, early, influential interactive languages.

• There are several videos showing demonstrations of historically significant systems: A two-tape video of the ACM Conference on the History of Personal Workstations is available from ACM, 212-869-7440, acm@acm.org. In addition to the presentations, early footage of the work of Engelbart, Kay, and Sutherland are shown.

ACM SIGGRAPH Video Review 12 & 13 has a demonstration of Sutherland's Sketchpad along with discussion of other work at MIT. It is available from First Priority, 800-523-5503. Alan Kay gives a lecture on his work on the tape "Doing with Images Makes Symbols: Communicating with Computers," available from University Video Communication, Stanford, Calif., 415-327-0131. In the course of his lecture, Kay shows demos of his, Engelbart, and Sutherland's systems.

ACM sponsored "The Machine that Changed the World," the 1993 Peabody Award-winning PBS series on the history of the computer. Many of the systems reviewed here are shown on the tapes, available from Films for the Humanities and Sciences, 609-275-1400.

• If you find this history interesting, read some of the original papers for fun, perspective, ideas, provocative quotes, humility, and to pay respect to the pioneers. The following are brief descriptions and a list of "modern" concepts and inventions found in several papers I would recommend:

Bush's article [3] influenced Engelbart and Kay, and it has been reprinted in many places, including [18]: voice synthesis and recognition, OCR, automated deduction, point of sale terminals and inventory systems, automated information retrieval, electroencephalographic input, magnetic storage and hyertext.

A report [4] includes a definition of Engelbart's theoretical framework, describes Bush's thought, and gives a detailed description of a hypothetical machine. It anticipates the field of computer supported cooperative work. Other papers [5, 6] describe the work which followed; cursor, word processing, hierarchial file storage, electronic meeting room, collaborative writing, outline processing.

An illustrated report [12] presenting Kay's vision, the software they developed, and the use children made of it. A subset of this report, with many illustrations, was reprinted in the March, 1977 issue of *Computer Magazine:* overlapping windows, multifont, WYSIWYG editing, painting, animation, waveform design, musical score input and editing, turtle geometry, simulation, and tool building.

Licklider [13] stated the advantages of interactive computing, and anticipated the field of human-computer interaction, interactive computing, scientific visualization, pen-based computing, voice I/O, read-only storage devices, on-line libraries, wide-area networks, artificial intelligence, and wall displays.

Licklider [14] also envisioned the important role of computers as communication devices, and anticipated the Net and the field of computer-supported cooperative work: communication for the articulation and reconciliation of mental models, computer-based meeting rooms, the artificial separation of communication and computing, graphic as well as text and numeric data, computer-mediated communication, packet-switched, wide-area networks, programmable, adaptive agents, the economics of transportation-communication tradeoffs, access equity, security considerations.

Marill and Roberts [16] unveiled the plans for the ARPANET: remote login, file transfer, remote procedure calls, early packet transmission experiment.

Metcalfe and Boggs [17] described the original Ethernet: this report was reprinted in *Communications, 19,* 7, July, 1976. LAN, Ethernet protocols and topology, client-server computing.

Strachey's [22] proposal for multiprogramming, coined the term "timesharing": statement of asynchronous peripherals, maskable, prioritized interrupts, memory protection, supervisor-only instructions.

Sutherland [23] introduced direct manipulation, CAD, computer graphics, object-oriented software, cursor, window, clipping, rubberband drawing, a constraint-based drawing, gesture recognition, resizing, dragging, dynamic memory allocation, pen "gravity," variable display intensity for feedback, graphic object data structures.

Thacker, et al. [24] describes the hardware, firmware, and software architecture and utilization of the first modern personal computer: client-server computing, Ethernet LAN, LAN-based internets, firmware support for windows and BitBlt, programmable cursor, modern editing, draw, paint, and page layout software, laser printer. ■

References

1. Albrecht, R.L. A modern-day medicine show. *Datamation,* (July, 1963) 31–33.
2. Baran, P., Boehm, S. and Smith, J.W. On Distributed Communications (11 Memoranda), RAND Corporation, Santa Monica, Calif., Aug., 1964.
3. Bush, V. As we may think. *Atlantic Mon. 176,* 1 (1945), 641–649, reprinted in [18].
4. Engelbart, D.C. Augmenting human intellect: A conceptual framework. Research Report AFOSR-3223, Stanford Research Institute, Menlo Park, Calif., Oct. 1962.
5. Engelbart, D.C. and English, W.K. A research center for augmenting human intellect. In *Proceedings of the 1968 Fall Joint Computer Conference.* Thompson Book Co., Washington, DC, pp. 395–410.
6. Engelbart, D.C., Watson, R.W. and Norton, J.C. The augmented knowledge workshop. In *Proceedings of the 1973 AFIPS National Computer Conference,* pp. 9–21.
7. Goldberg, A. *A History of Personal Workstations.* Addison-Wesley, Reading, Mass., 1988.
8. Kay, A. The reactive engine. Doctoral dissertation, University of Utah, Sept., 1968 (university microfilms).
9. Kay, A. Personal computing. Invited Paper, Meeting on 20 Years of Computer Science, Instituto di

Elaborazione della Infromazione, Pisa, Italy, June 12, 1975.
10. Kemeny, J. and Kurtz, T.E. Dartmouth time sharing. *Science, 162,* 3850, (Oct. 11, 1986), 223–228.
11. Kurtz, T.E. BASIC. In [25], 515–549.
12. Learning Research Group. Personal dynamic media. SSL-76-1, Xerox Palo Alto Research Center, Palo Alto, Calif., 1976.
13. Licklider, J.C.R. Man-computer symbiosis. *IRE Trans. Human Factors Electron.* (Mar., 1960), 4–11.
14. Licklider, J.C.R. and Taylor, R.W. The computer as communication device. *Sci. Tech.* (Apr., 1968), 21–31.
15. McCarthy, J. Programs with common sense. D.V. Blake, and A.M. Utley, Eds., In *Proceedings of the Teddington Conference on the Mechnisation of Thought Processes.* Her Majesty's Stationary Office, London, reprinted in M. Minsky, Ed., *Semantic Information Processing.* MIT Press, Cambridge, Mass., 1960.
16. Marill, T. and Roberts, L.G. Toward a cooperative network of time-shared computers. In *Proceedings of the 1966 Fall Joint Computer Conference,* pp. 425–431.
17. Metcalfe, R.M. and Boggs, D.R. Ethernet: Distributed packet switching for local computer networks. CSL-80-2, Xerox Palo Alto Research Center, Palo Alto, Calif., 1980. Also appeared in *Commun. ACM 19,* 7, (Jul., 1976).
18. Nyce, J.M. and Kahn, P. *From Memex to Hypertext: Vannevar Bush and the Mind's Machine.* Academic Press, Harcourt Brace Jovanovich, Boston, 1991.
19. Papert, S. Teaching children about thinking. IFIP Conference on Computer Education, 1970, Amsterdam, North Holland, pp I: 73-8.
20. Roberts, L.G. and Wessler, B.D. Computer network development to achieve resource sharing. In *Proceedings of the 1970 Spring Joint Computer Conference,* pp. 543–549.
21. Roberts, E.H. and Yates, W. ALTAIR 8800. *Pop. Electron.* (Jan., 1975), 33–38.
22. Strachey, C. Time sharing in large, fast computers. In *Proceedings of the International Conference on Information Processing,* UNESCO, Butterworth's, London, Jun., 1959, pp. 336–341.
23. Sutherland, I. Sketchpad: A man-machine graphical communication system. In *Proceedings of the Spring Joint Computer Conference,* (1963), pp. 329–346.
24. Thacker, C.P., et al. Alto: A personal computer. CSL-79-11, Xerox Palo Alto Research Center, Palo Alto, Calif., 1979. Reprinted in Siewiorek, Bell, and Newell. *Computer Structures: Principles and Examples.* McGraw-Hill, New York.
25. Wexelblat, R.L. *History of Programming Languages.* Academic Press, New York, 1981.

Larry Press is a professor of computer information systems at California State University at Dominguez Hills. He welcomes feedback and comments at lpress@isi.edu or 310-475-6515.