Fumbling the Future

How Xerox Invented, Then Ignored, the First Personal Computer

Douglas K. Smith and
Robert C. Alexander

toExcel
New York San Jose Lincoln Shanghai

Here is a three-part trivia question about televised personal computer advertising:

Name the companies responsible for

1. The longest playing series of personal computer commercials?
2. The most creative single commercial?
3. The first personal computer commercial?

Answering part one is easy. IBM's "Charlie Chaplin" ads ran for more than six years. They were entertaining, effective, and nearly impossible to avoid. Identifying Apple as the maker of the most creative commercial may be more challenging. Apple showed the ad just once, during the second half of the 1984 Super Bowl. Nonetheless, some people consider it the most impressive corporate identity commercial in history. Now for the last piece of the puzzle. Who televised the first personal computer commercial? This is not a trick question. It wasn't IBM, and it wasn't Apple.

It was Xerox.

Xerox is not a name most personal computer consumers, let alone general television audiences, associate with the multibillion dollar personal computing industry. Fifteen years after it invented the world's first personal computing system, and long after it portrayed that system in a 1979 commercial, Xerox still means "copy" to most people. Had it succeeded in marketing the computers shown in the commercial, however, Xerox might have meant more than copiers—much more.

Unlike Xerox, IBM, of course, always has been synonymous with computers. By far the most dominant personal computer advertising promotes the IBM PC. In it, a contemporary actor plays Charlie Chaplin playing his renowned tramp. The little man with derby, moustache, baggy trousers, and awkward walk twitters and jerks his way through the delightful discovery that computers can be useful and even fun for real people. IBM has spent massively on the campaign, as much to build interest in personal computing itself as to identify IBM's product as the standard in the industry.

In contrast to the IBM barrage, the memorable Apple commercial was more like a proclamation. Less than a decade after being incorporated in the garage workshop of two kids in their twenties, Apple Computer stood out as the Fortune 500 corporation best positioned to challenge IBM's dominance in personal computing. The brash, young California company selected 1984 and the Super Bowl to broadcast its commercial, a video morality play celebrating the glory of iconoclastic individualism and condemning the sinister threat of organizations whose power oppresses rather than liberates the human spirit. Using imagery without words, Apple drew the battle line clearly between itself and IBM.

There might have been a third competitor. In 1973, more than three years before Steve Wozniak of Apple soldered together a circuit board that qualified as a computer in name only, researchers at Xerox's Palo Alto Research Center (PARC) flipped the switch on the Alto, the first computer ever designed and built for the dedicated use of a single person. Long before Wozniak, prodded by his friend and partner Steve Jobs, went on to build his second computer—the famous Apple II, credited with changing forever the American home and workplace—and even longer before IBM implemented a crash strategy for breaking into and then dominating the personal computer industry, Xerox employees ranging from scientists to secretaries were using personal computers that, in many respects, were superior to any system sold in the market before 1984, the year of the Apple Super Bowl commercial.

The scientists at Xerox PARC created more than a personal computer. They designed, built, and used a complete system of hardware and software that fundamentally altered the nature of computing itself. Along the way, an impressive list of digital "firsts" came out of PARC. In addition to the Alto computer, PARC inventors made the first graphics-oriented monitor, the first handheld "mouse" inputting device simple enough for a child, the first word processing program for nonexpert users, the first local area communications network, the first object-oriented programming language, and the first laser printer.

They called this entirely new approach to computing "personal distributed computing." Their design and philosophy challenged accepted wisdom about the relationship between people and digital processors. Mainline computer people scoffed at the notion of one computer for each person; the Xerox team built the Alto. Traditional computer applications centered on number and data manipulation; the Xerox team focused on words, design, and communications. By the mid-seventies, PARC had crafted a framework of machines and programs that were "personal" because they were individually controlled, and "distributed" because they were linked through networks to shared resources and knowledge. The entire system—of people, machines, and programs—advanced human productivity through computing tools in ways paralleled only by the exploitation of pencil, paper, printing press, and telephone.

Xerox, however, did not convert either the vision or the implementation of personal distributed computing into the commercial success and recognition now enjoyed by Apple and IBM. It's not that Xerox failed to profit financially from its innovative technology. The company's laser printer business is thriving, and its latest generation of copiers incorporates technology developed at PARC. But these successes related easily to the world of imaging well-known at Xerox. By comparison, the greater possibility to define and dominate the unfamiliar business of personal computing smoldered unproductively within the company for more than a decade, frustrating far more of the organization than it inspired.

The Alto confronted Xerox with the unknown. When Xerox established PARC in 1970, there was no market for personal computers. There were no compact disc players, no Walkmen, no portable telephones, no digital watches, no VCRs, no video camcorders, no personal copiers. Not even the now ubiquitous pocket calculator had been introduced yet to the marketplace. Furthermore, from the time of its invention in the late 1940s through the end of the 1970s, computer technology remained unaffordable, inaccessible, and useless to most people. Computers were owned by corporations and universities, not individuals; operating the technology required a knowledge of protocols as formalized and arcane as any used in international diplomacy; and, all the effort yielded results for a narrow set of applications. For the most part, computers manipulated numbers in ways and with speeds helpful only to scientists, engineers, and accountants. Not surprisingly, popular films and novels depicted the technology as enigmatic and those who understood it as weird.

Except for the perception, all of this had changed by the time IBM introduced its personal computer in 1981. Consequently, IBM emphasized consumer education in its marketing strategy. If the Charlie Chaplin tramp could own a PC, the machine must be affordable. If he could operate one, the technology must be accessible. And if he could use a computer to better himself commercially and, yes, even romantically, then it must be useful.

The campaign was a remarkable success. By 1987, Americans had purchased more than twenty-five million personal computers. The machines were owned by one of every six households, and their absence in an office was far more remarkable than their presence. Children considered the technology routine. IBM's name was so identified with personal computing that IBM PC knockoffs, known as "clones," were grabbing a big share of the market for their United States and Asian manufacturers—so big that IBM ultimately changed its advertising strategy. The Charlie Chaplin character began touting the uniqueness of IBM products instead of merely demonstrating the wonders of personal computing in general.

IBM's early promotions made sense for a number of reasons. First, people did not have to be sold on the idea that IBM could make a good computer. Next, since IBM was the only personal computer manufacturer in the early 1980s willing and able to advertise extensively on television, it had no competition for what advertisers call the "share of voice." Television viewers simply didn't see or hear that much about the competition. As a result, IBM could educate consumers while relying on sheer omnipresence to associate its product with a safe and wise choice. Finally, the approach succeeded because, by 1981, enough personal computer hardware and software was available in the marketplace to back up the discovery claims made by IBM's little tramp.

Only five years earlier that had not been the case. The first personal computing products appeared in the mid-seventies and had limited appeal. They were sold by small electronics firms and individual hobbyists through clubs, direct mail, and word of mouth to other hobbyists and tinkerers. Wozniak's Apple I typified the early merchandise. It was an unpackaged circuit board wired by Wozniak so that a purchaser could hook it up to a power supply

(not included), connect a tape cassette for input (not included), a television for output (not included), and then set about writing programs (not included) to fit within the Apple I's limited internal memory. Millions of Americans preferred spending their time in other ways.

Within a few years, however, astonishing advances in integrated circuitry provided the critical raw materials needed by hobbyists and others to build bigger, better, and more useful computing tools. Personal computer memories, speed, and power expanded. Disc drives, keyboards, mice, monitors, and printers were added. And, most important, programmers began writing routines to make the machines appealing to people other than tinkerers.

At first, many programmers focused on games. But by 1979, data base management, word processing, and the electronic spreadsheet all had been invented. With the emergence of these applications, large numbers of people realized that the small computers could help them manage information more productively, write and type better, and think more clearly. The personal computing market, having rung up its first sale in 1975, measured revenues in the billions of dollars by 1981.

Few opportunities have ever burst onto the scene so suddenly and with such force. To thrive on the shock of such an explosion required not only good, responsive products but the faith and hustle to profit from them. Apple had that magic combination. Theirs was the classic American business story starring two high school graduates with little money, no economic training, and big dreams. Wozniak built and improved the product; Jobs provided the faith and the hustle. When Jobs's energy exceeded his understanding, he recruited more experienced manufacturing, marketing, and financial managers to guide Apple through its rapid expansion.

By 1984, the year George Orwell predicted would witness a tyranny of computers in the hands of evil men, Apple Computer, like the personal computing industry at large, held out the opposite promise. Apple marked the event with its Super Bowl commercial. The ad begins with several indistinguishable cohorts of gray-clad ideological slaves marching in lockstep toward a great hall. Once inside, they take instruction from a larger-than-life image projected on a screen at the head of the auditorium. In the midst of this lifeless, impersonal scene, a powerfully built woman, dressed in bright colors and wielding a sledgehammer, charges into the hall and spins herself around and around and around, frightening the brainwashed masses. With each of her revolutions, the tension grows in the great hall until, finally, at the end of the piece, she launches her weapon directly at the big screen.

The commercial's imagery richly conveyed Apple's perspective on its history, computers, and IBM. Perhaps more subtly, the television time purchased told as much about Apple the corporation. Super Bowl minutes are the most expensive advertising time in the world. Apple may have had an antiestablishment past, but its economic power in 1984 was as conventional and formidable as the beer, car, and financial services companies who also sponsored the annual football championship. The Super Bowl spot marked Apple's arrival; it was only the second company in history to have reached a billion dollars in sales in less than ten years on the merits of a new technology.

The first was Xerox. Less than a decade after the 1959 introduction of its revolutionary office copier, Xerox went over the billion dollar mark and claimed a position, along with IBM, as one of America's leading office products companies. By 1970, competition between the two giants seemed inevitable as each rushed into the technology of the other—IBM into copiers, and Xerox into computers. At the time, business computers were stationed in corporate back offices, handling the work of accountants and statisticians. No one expected them to stay there. So, in addition to taking on IBM in back office computing, Xerox established its Palo Alto Research Center to invent systems that could support executives, secretaries, salesmen, and production managers in what became known as the "office of the future."

The remarkable group of scientists and engineers who joined PARC responded with the Alto personal distributed computing system. Xerox's 1979 commercial demonstrates how the Alto functioned in an office setting. We see friendly "Bill," a balding middle-aged executive with a warm smile, arrive at work, grab a cup of coffee, and head for his office, saying good morning to people on the way. When Bill gets to his desk, he flips on his Alto computer, grins, and greets it with a "Morning, Fred." "Fred" the computer flashes the appropriate response: "Good morning, Bill."

Bill asks, "What's the mail this morning?" and then scans a list showing the times and origins of messages he has received since leaving work the day before. "This one looks interesting," says Bill. "Let's, ah, take a look at this." He selects the desired message

with the aid of his mouse, and the full text fills one section of Fred's monitor.

After reading it, Bill tells Fred, "I'm going to need a couple of copies of this." Bill presses a button that controls an off-camera laser printer, and the commercial cuts to some time later when a secretary delivers Bill the paper copies he's requested from the printer. He thanks the secretary, then turns back to the computer saying, "Oh, and thank you, Fred. You know, Fred, I think everyone on the routing list should see this." So Bill pushes a few more buttons, sending electronic copies of the message down the hall, around the corner, and across the country.

The commercial highlights many parts of the Xerox system including the graphics-rich Alto screen, the mouse, the word processing program, the laser printer, and, most prominently, the system's communications capabilities. It's an effective ad—other than the Xerox name, nothing about it would surprise a television audience even if it were shown today.

But in 1979, despite airing the spot several times, Xerox decided against marketing the Alto system. By then the organization barely resembled the buoyant company that a decade earlier had challenged both IBM and the office of the future. External factors including fierce competition, government antagonism, and economic recession all marked Xerox's slide—from overconfidence to loss of confidence. Internal forces were even more combustible, as the company's research, finance, and marketing groups each pursued a separate vision of the "right" Xerox future. In the end, the company that invented the first version of a personal computing future found itself struggling to recapture the advantages of its copier past.

In one fundamental respect, neither economists nor business people would consider the corporate histories behind the three different personal computer commercials that remarkable. Of course IBM waited for personal computers to move beyond hobbyist circles before entering and dominating the market. It's a well known strategy for firms with established economic power to take advantage of the innovation and product testing done by others. Of course a start-up like Apple flourished. Rags-to-riches entrepreneurs are among the most cherished citizens in capitalist economies. Of course Xerox stumbled.

But why? Why do corporations find it so difficult to replicate earlier successes in new and unrelated fields? How could Xerox, sired by one radical technology, bring forth yet another extraordinary invention, only to fumble away most of the economic opportunity it promised? It doesn't have to happen this way. One clue to why it did happen to Xerox, and why it's now occurring but shouldn't be at other corporations, is found in the conclusion of the Alto commercial. We cut to quitting time for a final dialogue between "Bill" and "Fred" the computer:

> Bill (tired): "Anything else?"
> Fred: A richly detailed bouquet of daisies spreads across the screen.
> Bill (puzzled): "Flowers? What flowers?"
> Fred: "Your anniversary is tonight."
> Bill (chagrined): "My anniversary. I forgot."
> Fred: "It's okay. We're only human."

Marketing: The Architecture of Information

Chapter 7

In designing and building the Alto, Butler Lampson and Chuck Thacker had to master two conflicting objectives—how to make a system *cheaper and better* than minicomputers. Unless the cost of an Alto fell considerably below the price of a minicomputer, PARC would not have been attracted to the notion of replacing timesharing with an experiment in personal computing. Yet, as Lampson and Thacker conceived it, "personal" implied convenience in addition to economy. To them, a personal computer had to handle as easily as other common instruments of expression and communication—typewriters, blackboard pointers, pencils, pens, and paper. Only a state of the art system could approximate that kind of functionality, but only a stripped down computer would be affordable. Only an elegant solution would do both.

"Timesharing systems," according to Thacker, "had made computing more accessible and decreased its cost, but they had done little to increase the *quality* of man-machine interaction." Programmers drew charts and pictures with bulky characters like *1*'s and *x*'s rather than fine lines and points. Most people viewed their output on teletype machines instead of display terminals, and those lucky enough to have monitors suffered eye strain from low grade video performance. Documents looked very different on the user's display from when finally printed out; printouts themselves, on oversize paper with barely legible type, were no joy to read.

Lampson and Thacker had seen better "user interfaces," especially in the research of Douglas C. Engelbart, one of the patriarchs of interactive computing. In the 1950s, when many people feared computers might someday control or, worse, replace human beings, Engelbart originated a contrary view. Digital systems, he argued, should augment human intelligence, not automate it. He waged a lonely campaign for augmentation until 1964, when Bob Taylor, then at NASA, funded him to build his own research laboratory. Four years later, in a seminal event in the history of computing, Engelbart demonstrated an unprecedented variety of interactive technology. He introduced a national conference of computer scientists and engineers to, among other things, a hand-held input device called a "mouse," television monitors that could be divided into multiple "windows," and software with powerful outlining features to facilitate structured thinking and presentation.

Engelbart's mouse was an analog device housing large steel wheels and a series of buttons. The motion of its wheels controlled the cursor, the highlighted marker on a computer screen that indicates the current position of interest to the user. By rolling the mouse over a flat surface, people could move quickly through their work, pointing the cursor at target areas, and clicking the buttons to enter commands.

For the Alto, Lampson and Thacker hired an inventor to convert the mouse into a digital device, reduce its size, and simplify its handling and reliability. They also commissioned themselves to improve Engelbart's approach to television displays, hoping to engineer the Alto's screen to simulate the familiarity and flexibility of ink and paper.

Think for a moment about the infinite possibilities of these two simple tools. Ink renders characters of any size, shape, and style; it draws straight lines and curves; it produces textures and halftones. Ink can be positioned anywhere on paper with a high degree of resolution. Sheets of paper can be spread out and worked with concurrently, or bound together and reviewed a page at a time. Variation, amount, complexity, structure, relationship—the basic aspects of information are captured and communicated with ink and paper.

"Only one technique," Thacker once noted in a flurry of jargon, "is known for approximating all these properties in a computer-generated medium: a raster display in which the value of each picture element is independently stored as an element in a two-dimensional array called a bitmap."

This can be explained.

On average, Americans spend more than seven hours every day in front of raster displays. They call it television. A television monitor is, in effect, a grid of dots, or "picture elements," made by dividing the screen horizontally and vertically. Each picture element is like a microscopic light bulb. Images are formed when an electron beam scans back and forth across the screen, illuminating the picture elements with intensities that vary as a function

of the originating television camera. "Raster" refers to the pattern of horizontal scanning.

Television is an analog technology. Nonetheless, the electron beam and picture elements provide a perfect digital application. A computer can turn the beam on or off; it can store each picture element's on or off status in memory. For example, this letter *A* might occupy a rectangle of four by six picture elements. If the electron beam lights up the correct picture elements, an *A* will appear against the background of the screen. By associating each picture element with a specific bit of computer memory and programming the appropriate bits to be "on" while the others are "off," the user both creates an *A* on the screen and reserves it in memory for future recall, as follows:

Computer Memory	**Computer Screen**
0110	11
1001	1 1
1001	1 1
1111	1111
1001	1 1
1001	1 1

Lampson and Thacker called this technique "bit mapping" because of the one-to-one correspondence between the bits in the computer's memory and the picture elements on the screen. How well they could capture the desired properties of ink—characters, lines, curves, textures, halftones, and positioning—depended upon the size of the picture elements. The smaller, the better. For example, two hundred picture elements per inch would produce video images of extraordinary richness.

That quality, however, came at a steep price. The Alto's screen re-created the standard 8½- by 11-inch piece of paper. (This was done by taking a TV screen, normally wider than it is tall, and turning it on its side.) Thus, the display area approximated 100 square inches. At two hundred picture elements per inch in each direction, the Alto would need nearly four million bits of memory for its bit map. In 1972, installed bits of memory cost just under one and a half cents apiece; a four-million-bit screen would have run more than $50,000, far too much for a personal computer.

Other display technologies required less memory. The most common was the calligraphic display. It too employed an electron beam and a fluorescent screen, but it operated the beam in a different manner from a raster television monitor. Instead of illuminating a grid of picture elements, a calligraphic electron beam painted patterns in continuous motions like brush strokes. The computer stored the location of the first and last points to be lit up; the electron beam completed the line. Thus, for any given stroke, a calligraphic display required only a handful of bits to govern where to start and where to stop.

But calligraphic displays had a major defect. They flickered. Screen images made by electron beams are not permanent. That's why the television screen goes blank when you switch it, and the electron beam, off. In fact, televised patterns must be illuminated at least thirty times a second to appear fixed; anything less produces a disturbing instability. When calligraphic technology portrayed a complex image like this page of text, the time required to trace each character prevented updating the screen often enough. Because raster technology controlled the computer image completely with switches (turning the beam on and off) instead of relying on brush strokes (drawing with the beam), it could re-light the screen much faster. Raster images appeared stable; calligraphic ones did not.

Consequently, only a bit mapped raster display, notwithstanding its voracious appetite for memory, could capture the richness of ink and paper while eliminating flicker. "Fortunately," Lampson and Thacker discovered, "surprisingly good images can be made with many fewer bits." They believed a grid of 500,000 picture elements instead of 4 million would "preserve the recognizable characteristics of paper and ink." Still, when finished, the Alto's bit map consumed nearly half of the computer's total memory. While Lampson, Thacker, and others could, and did, develop techniques to cut back on the display's demand for memory, the resources willingly dedicated to the Alto's user interface remained impressive.

The improved mouse and bit map display augured well for PARC—*if* Altos could be built inexpensively enough for everyone to have their own. Thacker accurately predicted the Alto's total memory would cost only $35 dollars in the early 1980s. But in 1972, it went for $7,000. He and Lampson had to find other ways to save money.

In one respect, the choice of a raster display helped because large scale television manufacturing made raster monitors cheap to purchase. Lampson and Thacker also economized by stripping the Alto's arithmetic functions to a bare minimum. And the abandonment of timesharing yielded up some savings since protecting users against one another became unnecessary. Finally, Lampson and Thacker knew PARC would assemble its own machines, eliminating the expense of labor, administrative overhead, and profit included in the price of another company's products.

All these measures combined, however, were insufficient to meet the Alto's cost objective. Minicomputer performance specified more hardware than the Alto budget could tolerate. Moreover, in light of the history of computer engineering, sophisticated facilities like the bit mapped display should have added, not reduced, the circuitry in the system.

Computers consist of four major parts: input, central processor, memory, and output. Data and instructions are transmitted from an input device such as a keyboard to memory. The central processor fetches the data and instructions from memory, and it executes the required computation. The central processor then dispatches the results to an output mechanism like a screen or a printer.

This scheme has an added wrinkle. The central processor itself comprises two subunits. Its "arithmetic and logic unit" manipulates the data to produce computing results; its "control unit" keeps order throughout the system, much as an air traffic controller directs takeoffs and landings to prevent collisions.

Control units in first generation computers were required to direct more traffic than is typical today. They did virtually all of the electronic processing necessary to operate each of the computer's other subsystems: arithmetic and logic, memory, input, and output. This slowed down processing.

Recall that the central processor carries out a discrete step with each separate clock beat, or cycle. More work means more steps; more steps mean more cycles; more cycles, or clock beats, mean more time to completion. By burdening the central processor with the jobs of input and output as well as arithmetic, logic, and memory, computing results took longer to obtain. Furthermore, to minimize the demand for cycles, input and output devices had to remain relatively primitive.

Engineers tackled this dilemma during the 1950s and 1960s by adding processing circuitry directly to input and output accessories. Successive layers of hardware assumed more and more of the workload until, in the most advanced systems, input and output devices incorporated their own processors and memory, freeing the central processor's control unit of all responsibilities other than general coordination. Since the central processor could devote more cycles to memory, arithmetic, and logic, and less to input and output, computer systems ran faster. In addition, the added circuitry paved the way for more advanced input and output technology including keyboards, disk drives, and displays.

Of course the extra circuitry also cost a lot of money. Chuck Thacker realized a return to the concept of sharing a processor's cycles with input and output would reduce the parts bill for an Alto. The dilemma was how to cut back on hardware without sacrificing features like the bit map display that required access to powerful electronics—in other words, how to subtract circuitry while adding capability. A neat trick.

Thacker says, "The solution just came to me. It was an 'ah ha' experience."

His innovation, called "multitasking," effectively turned one processor into many. He wired the control unit of the Alto's central processor to take its instructions from up to sixteen different sources, or "tasks," instead of the usual one. Among these tasks were the bit map display, the mouse, the disk drive, the communications subsystem, and the user's program. The tasks were assigned priorities: if two or more of them signaled a request, the one with the highest rank took possession of the processor. When the display was in control, it was the display's processor; when the disk drive had precedence, it was the disk drive's processor; when the mouse took charge, it was the mouse's processor; and so on.

The instructions themselves controlled traffic. Each instruction contained information about its successor. Say the user's program was in control of the processor. As each instruction was processed, the ensuing instruction was fetched automatically from memory and signaled a request to continue using the processor. If this next instruction still had top priority, it too got processed, and the pattern was repeated. But when an instruction for the user's program ran into competition, for example, from the higher ranking bit map display, the user's instruction moved to an electronic warehouse while the display's instruction took hold of the processor. The user's instruction stayed in the warehouse until its

request for the processor once again held the highest priority. At that point, the computation of the user's program continued exactly where it had left off.

Instructions for all sixteen tasks were subject to the same rules. If they had priority, they got processed; if not, they were maintained in the warehouse. As a result, the instructions for the display, disk drives, mouse, user's program, and other tasks were executed in the proper order regardless of when, and for how many consecutive cycles, they had control of the processor.

Multitasking provided more functionality for less cost. Priority control of the Alto's powerful central processor meant the computer's input and output facilities could perform sophisticated feats without their own circuitry. Total system hardware requirements dropped by a factor of ten. The parts bill for an Alto ran just over $10,000, about 60 percent less than spent on the components for a minicomputer.

Multitasking did slow down the Alto. The bit mapped display controlled the processor two-thirds of the time, leaving the rest of the system just one out of every three clock cycles to complete its work. Therefore, instructions and data took three times longer than normal to compute. The delays, however, were measured in microseconds. Furthermore, unlike timesharing, the speed of the Alto was thoroughly predictable. As one of PARC's researchers would later declare to the general applause of his colleagues, "The great thing about the Alto is that it doesn't run faster at night."

In November of 1972, with the Alto's design in hand, Thacker went to work on a prototype. He was joined by Ed McCreight and Larry Clark. Thacker says, "The hardware business is not like software. In software, you get immediate feedback. In hardware, there is a fairly long, unkind period when you have no idea if this pile of junk is going to work!"

By that measure, assembling the first Alto was exceedingly kind. "It worked just the way it was supposed to," recalls Thacker. "It was the most satisfying hardware system I had done, working essentially the first time. And that was because its design was so simple."

It took under four months to build the system. The first picture displayed on the new bit mapped screen was the Sesame Street Cookie Monster, which had been programmed as a test pattern by a member of Alan Kay's group. Says McCreight, "I hadn't really imagined what the bit map display would be like until I saw it running Alan's Cookie Monster. He had digitized two frames, and by switching back and forth, we got to see the Cookie Monster eating a cookie."

Thacker, too, was thrilled. "I remember checking out the display. It was late at night. There were only the three of us standing around and we were, of course, overjoyed. The display worked. You could see it! We all knew intellectually that this thing was going to be neat. But until it worked, we hadn't truly internalized what it would mean to be able to put pictures on the screen and change them on the fly. That made it all much more real. Later, people would walk by, and we'd just point, like the proud parents outside the maternity room in a hospital."

Chapter 8

"What really captured the imagination of most computer scientists in the lab was the Alto," said one PARC engineer. "Two things about it were very different from before. One is that you had your own computer. And that means you're willing to use it for a much broader range of tasks because you don't sit there continually worrying about 'Gee, should I really be doing this or am I using up cycles that somebody else could be using to better advantage?' The second thing was the whole technology of the mouse and the bit map display. It was just a prod to your imagination. How could I exploit this wonderful new feature? What are the new things we can build out of this?"

There were many, many new things to build. The Alto may have been the world's first personal computer, but hardware alone does not a computing system make. Without software ranging from operating systems to programming languages to application packages, the Alto, in Chuck Thacker's words, "was no better than a hot rock—interesting but useless."

Thacker's team finished the first Alto in April of 1973. By the end of the year, PARC had ten Altos; by the following summer, the lab had forty. As the machines spread and the basic enabling software fell into place, dozens of projects were initiated. Of these, three particular innovations—in communications, printing, and word processing—illustrate how the Computer Science Laboratory, often aided by researchers from the Systems Science Laboratory, employed their Altos to advance the state of the computing art.

"In a few years, men will be able to communicate more effectively through a machine than face to face." When Bob Taylor had made that prediction in a 1968 paper coauthored with J.C.R. Licklider and Evan Herbert, computer-to-computer communications barely existed. Programmers sharing the same computer in a timesharing system could exchange information, but users of different computers could not. The difficulty, Taylor, Licklider, and Herbert had noted with disappointment, was that the nation's multibillion dollar telephone system left data communications out in the cold.

The telephone network operated in a pattern known as "circuit switching." As a phone call was dialed, telephone company operators and equipment made a series of connections from one node in the network to the next until a full circuit ran between caller and receiver. It took a second or two to complete all the parts of a circuit, but nobody much cared. People tended to talk on the phone for several minutes; a few seconds' wait was no bother. For computer communications, however, the set-up time was an eternity. Unlike telephone conversations, data exchanges lasted microseconds, not minutes. And a meaningful data message often required dozens, even hundreds, of separate transmissions. Interposing a second or two between each burst of digital code would have caused excruciating delays.

It also would have been prohibitively expensive. In order to amortize the heavy cost of circuit switching, the phone company charged its highest rates for the first minute of conversation even if the connection lasted a shorter time. Imagine the phone bill of a Boston programmer who incurred the full first minute charge for each one of the thousands of keystrokes needed to enter data and instructions into a California computer. The invoice would have extended for pages; its total would have exceeded the programmer's salary.

In the late 1960s, a handful of computer scientists developed an alternative to circuit switching, called message switching, which led to the ARPAnet, the first nationwide computer network. In message switching, information was routed through the ARPAnet like a baton in a relay race. Data traveling from A to E made the full trip in steps from A to B, B to C, C to D, and, finally, from D to E. This protocol was known as "store and forward." The ARPAnet computer at the first center waited for a clear line, then transmitted its message to the second center. Meanwhile, the first center's computer stored a copy of the message until the second center's computer acknowledged receipt. At that point, the second center's machine waited for a clear line, then forwarded the data on to the third. And so on, until the message reached its ultimate destination.

Since each of these computer-to-computer links happened in thousandths of a second, hundreds of digital dispatches could travel from origin to destination on the ARPAnet during the second or two it would have taken the telephone system's circuit switched network to even begin transmission. And because multiple node circuits were unnecessary, the bulk of the telephone system's expensive set-up charges were avoided.

The ARPAnet system worked well for long distance computer communications. But combining the Alto computers at PARC into an inexpensive local network posed a dilemma. Requiring a customized store and forward computer "partner" for each Alto in a PARC network would have destroyed the economics of personal computing. Unfortunately, without store and forward protocols, there was no way to prevent interference. Just as simultaneous telephone conversations on the same circuit are no good, two or more digital messages sent at the same time over the same wire collide, rendering their signals unrecognizable. That's why the telephone system has busy signals, and that's one reason the ARPAnet had store and forward computers.

Therein lay the challenge and the genius of PARC's short distance, or "local area network." With the help of Lampson and Thacker, Robert Metcalfe of CSL and David Boggs of SSL solved the reliability problem with a neat twist—they didn't insist on success. Their invention, called "Ethernet," plugged Altos into a cable strung throughout the building at PARC. Each Alto broadcast messages to the entire network but received just those dispatches carrying the correct address. Thus, unlike store and forward systems, the Ethernet permitted simultaneous transmissions and interference.

The lightning speed of a computer processor inspired Metcalfe and Boggs to tolerate failure. As long as the incidence of bollixed transmissions was kept to a minimum, they figured a blocked message now and then would not appreciably slow down overall communications. In their Ethernet system, if a transmitting Alto detected interference, it stopped broadcasting, waited a random number of microseconds, and tried again.

"The basic idea," says Butler Lampson, "is very, very simple. Imagine you're at a party and several people are standing around having a conversation. One person stops talking and somebody else wants to talk. Well, there's no guarantee that only one person wants to talk; perhaps several do. It's not uncommon for two people to start talking at once. But what typically happens? Usually they both stop, there's a bit of hesitation, and then one starts up again. That's essentially how the Ethernet works."

The Ethernet connected Altos to other Altos and to equipment shared by everyone in the laboratory. The most popular pooled resource was another PARC invention, the first xerographic laser printer. It combined the scientific legacy of Chester Carlson with the engineering wizardry of PARC's talented researchers. Xerography is a product of light and shadow. Put your hand up to a light, and the shadow of the hand appears on an opposite wall; put your hand on the glass of a Xerox copier, and the shadow of the hand appears on an electrostatically charged metal drum within the machine. The drum's charge is neutralized by light, perpetuated by darkness. Therefore, the metal surface remains charged only in that area of shadow corresponding to your hand. Black powder bearing an opposite charge is sprinkled over the drum. It sticks to the area of shadow and slides off the rest of the metal surface. A clean sheet of paper adheres to the powder, paper and powder are fused with heat, and a copy of your hand is set to roll out of the machine.

In the 1960s, a few engineers at Xerox's Webster, New York, research facility thought xerography might have applications beyond copying. Among them was Gary Starkweather, who demonstrated that lasers—powerful light sources—could create xerographic facsimile images. In 1969, Starkweather proposed to extend Carlson's process by inventing a laser printer. His boss said no. But his boss's boss, George White, believed lasers had great potential for Xerox, and he recommended that Starkweather transfer to PARC.

The computer scientists welcomed both Starkweather and his idea with enthusiasm. "They saw a common architecture in all of this," Starkweather remembers, "because they had been looking for something that could print in a bit map fashion."

If bits of computer memory could correspond to the picture elements of a television screen, they could also be mapped to dots created by a laser on a xerographic drum. In the printing application, the digital *1*'s and *0*'s control the path of the laser. If the laser illuminates a spot on the metal drum, its light neutralizes the electrostatic charge, the black powder slides off that tiny area of the drum, and the paper stays white. On the other hand, if the laser is prevented from reaching the drum, the spot stays dark, its electrostatic charge survives, the powder sticks, and the paper turns black.

The artistic method here is a hi-tech version of pointillism. Starkweather and the computer scientists conceived of the metal

drum as a grid with five hundred dots per inch in each direction, providing more than twenty-three million dots for marking a standard 8½- by 11-inch page.

The laser had to shine on and off the metal drum with fantastic speed. To build his printer, Starkweather modified a Xerox 7000 Copier, which operated at the rate of one page per second. Subtracting the time required for paper to travel through the machine left the laser about two-thirds of a second to scan the twenty-three million dots on the drum. That meant the laser had to be turned on and off nearly forty million times a second.

Starkweather produced the necessary effect with a magician's touch. To the light and shadow trick of xerography, he added sound and mirrors. First, he designed a twenty-four sided polygon about the size of a doughnut. Each facet held a mirror to reflect the laser beam. By mounting the doughnut on a revolving shaft, Starkweather made it possible to scan the entire drum. And with the aid of sound energy, he figured out how to bend the laser on and off rapidly enough to mark the twenty-three million separate spots in the time allowed.

All of which called for a good deal of engineering creativity. Starkweather's most ingenious solution, however, was to a more difficult problem. To print clear and precise images, the revolving mirrors had to bounce laser beams against the drum with great accuracy. Think back to the bit mapped letter *A* discussed in the previous chapter. Let each *1* represent a dot created by the laser. If the laser did not scan the drum exactly as intended, the *1*'s would appear unaligned, leaving wobbly images on the final copy, as follows:

Computer Memory	**Laser Printout**
0110	11
1001	1 1
1001	1 1
1111	111 1
1001	1 1
1001	1 1

To prevent uneven pictures, Starkweather calculated that the position of any dot could not vary more than one thousandth of an inch from its "theoretical center," or target. If each mirror on the polygon were not cut to perfection, or if the bearings in the motor driving the spinning shaft did not operate exactly as specified, the laser's angle of reflection from the mirror would carry it beyond the required tolerance, producing a distorted picture.

Fabricating near perfect mirrors and bearings was possible but far too expensive. Starkweather devised a much simpler and cheaper approach. He chose to employ optical means to correct errant lasers instead of fabrication and mechanical means to prevent the irregularities. By interposing a special kind of lens between the rotating mirrors and the xerographic drum, he managed to concentrate the laser on the drum exactly where intended. Light that bounced off the spinning mirrors heading in the wrong path was caught by the lens, corrected, and directed to its intended target on the drum.

Starkweather named his machine "SLOT," for "Scanned Laser Output Terminal." To use an automobile analogy, SLOT was a fast and fancy set of wheels. But it required an engine to drive it, that is, to tell it *when* to bend the laser toward or away from the xerographic drum. That "engine" was a digital processor and memory called the "Research Character Generator" invented by Ron Rider of SSL under the guidance of Butler Lampson.

History's best known "character generator" is Johannes Gutenberg, who invented movable type in the fifteenth century for his famous Bible. Over the next five hundred years, craftsmen contrived a spectrum of typeface styles and sizes. Gothic, Greek, Helvetica, Times Roman, light, medium, bold, italic, serif, sans serif—these and other terms describe the typesetting choices available to printers.

Gutenberg poured an alloy of lead, tin, and antimony into molds to cast separate characters that could be set in any arrangement prescribed by a manuscript. Lampson and Rider shaped bit patterns for exactly the same purpose. With patterns of *1*'s and *0*'s, they could signal the laser to print any letter from a Times Roman italic *a* to a Helvetica boldface *z*. In theory, their Research Character Generator could have captured the entire heritage of the printing trade. But in practice, the number of memory chips required to store so much diversity exceeded their budget. Nonetheless, even a partial collection of type styles and sizes would provide PARC researchers with a rich and varied digital printing shop.

Rider recalls confronting an unusual problem at the beginning of his project. Xerox copiers run pages horizontally instead of vertically. Compare this with a typewriter. Pages are inserted vertically, and typing is done horizontally, one letter at a time in the same direction as reading. The Xerox 7000 Copier Gary Starkweather modified into the Scanned Laser Output Terminal had a different orientation. Pages went through the machine horizontally. As a result, the laser made its black and white dots vertically.

"This caused problems for building an image generator," explains Rider, "because you come across all lines at once." Look at the following passage taken from a sign Bob Taylor hung outside his PARC office (proving the biblical origin of binary computer logic):

> But let your communication be,
> Yea, yea;
> Nay, nay:
> For whatsoever is more than these
> Cometh of evil.
>
> —*Matthew, Chapter 5, Verse 37*

To print this verse, SLOT's laser scanned from top to bottom, not right to left. The first scan left a series of black and white dots representing a vertical slice of the letters *B*, *Y*, *N*, *F*, and *C*. The Research Character Generator retrieved the correct pattern of *1*'s and *0*'s from its memory for each letter, communicated the bits to the laser in the proper order, and kept track of the results so that, on the next pass of the laser, the machine could pull out the appropriate follow-on bits and send them to the laser. This cycle repeated until the page was complete. In this manner, images were assembled like multidecker sandwiches, one page length slice at a time.

Rider gleefully describes these digital logistics as a "massive sorting kind of disaster." With Lampson's help, he completed the character generator by laying out and wiring nearly twenty-five hundred integrated chips. When he finished, PARC's remote control print shop was in business. For a technology that had everything to do with visualization, PARC's computer people dreamed up the anatomically strange acronym of "EARS"—for the *E*thernet, the *A*lto, the *R*esearch Character Generator, and the *S*canned Laser Output Terminal.

Everyone at PARC—scientists, administrators, and secretaries—used EARS to print elegant documents. Before they could print, however, they had to compose. And the word processing package employed most often was a program called "Bravo," designed by Butler Lampson and another CSL researcher, Charles Simonyi.

Bravo introduced computer scientists to a fresh concept called "wysiwyg" (pronounced "whiz-ee-wig") for "what you see is what you get." In Bravo, the appearance and layout of a document on the Alto screen were identical to the hard copy printed by EARS. By contrast, software predating Bravo forced computer users to interleave text with specific formatting instructions that commanded a piece of software known as a "document compiler" to convert the input into a normal document. Consequently, what people saw on their screen—the desired words and numbers interrupted by the necessary formatting commands—was not what got printed out when they were finished.

"The document compiler," explains Lampson, "played the same role as a human typesetter. Instead of giving a manuscript to a typesetter, you gave instructions to the document compiler. Typesetters don't need explicit instructions for formatting because they carry the rules in their head. But a document compiler has to be told everything."

The document compiler routine of having to include command instructions in the body of a composition was disruptive to programmers and beyond the grasp of the uninitiated. Eliminating it required Lampson and Simonyi to program Bravo to perform as a document compiler each time the user entered a change of text. This was quite difficult.

A document compiler calculates all of its layout information at once, a sensible scheme when users complete their writing before the formatting program goes to work. But in Bravo, formatting calculations were to be processed during the composition session, not after it. Therefore, the effect of every textual insertion and deletion, every margin adjustment, every underline, every superscript and subscript—in short, of every change—had to be computed immediately.

Say we wished to add a word to the next sentence in this paragraph. A document compiler would recompute the layout of the entire chapter, taking perhaps as long as a minute or so to finish. If every correction or alteration of text in Bravo had caused

such a lengthy delay, Lampson and Simonyi would have succeeded only in replacing one kind of interruption, the command insertions, with another—slow response times.

"The trick," says Lampson, "was to try to limit the propagation of changes as much as possible." They did this by raising digital walls. Bravo treated every line of text as though it were a separate document. When an insertion, deletion, or other change was made, the Bravo program, in effect, polled each line to find out which ones required alteration. When it recomputed, it did so only for the lines that mattered. As a result, the changes processed by Bravo appeared on the screen in a tiny fraction of a second. To the user, it looked instantaneous.

Bravo incorporated much of the new hardware and software at PARC. Text entries and changes were immediately visible on the Alto's bit map screen, a mouse could be used to point to selected portions of a document, windows were available to compare different sections of text, and results could be communicated over the Ethernet to other Altos as well as to EARS to get a printed copy that looked exactly like it was supposed to. Interactive computing had been extended to an entirely new frontier. The people at PARC who had Altos felt blessed—nowhere else in the world was there a computing system to match it.

Chapter 9

The first Xerox executive to seek an advantage from PARC's revolutionary technology was a man named, appropriately enough, Darwin Newton. Newton was the senior administrative editor of Ginn & Co., a Boston-based textbook publisher owned by Xerox. As such, his primary concern was productivity. In the early 1970s, newspaper, magazine, and book companies began experimenting with mainframes and minicomputers to expedite the publishing process, and Darwin Newton wanted PARC to help him improve efficiency at Ginn.

In 1974, Newton contacted Bill Gunning at PARC. After briefly serving as manager of the Systems Science Laboratory, Gunning had accepted an assignment from George Pake as technical liaison between PARC and the rest of Xerox. In response to Newton's inquiry, Gunning put the Ginn editor in touch with Bill English, the leader of an SSL project called "POLOS" for "PARC On-Line Office System." English, in turn, asked one of his researchers, Larry Tesler, if he would assist Ginn.

It was an ideal opportunity for Tesler. After graduating from Stanford in 1965, he had written software for computer-aided editing, formatting, and page layout. In addition, he'd nurtured an abiding interest in making computer tools more accessible to nonprogrammers. So the Ginn project appealed to both his interests and his inclination. And, as Tesler recounts it, the assignment also had a certain attraction to English and the rest of the POLOS team. "I didn't like the POLOS architecture at all," says Tesler, "and I complained about it a lot. I was disruptive. They were happy to move me to one side."

It soon became apparent, however, that adapting POLOS to Ginn's specifications, if possible at all, would be difficult for Tesler to do alone. To avoid diverting anyone else from the POLOS team, English proposed that Ginn hire its own computer person to help Tesler. English and Tesler wrote up a job description, had it posted throughout Xerox, and told Newton they would interview any worthwhile candidates.

In September, Newton introduced them to Tim Mott, a twenty-four-year-old Englishman who had come to the United States three years earlier with a computer science degree from the University of Manchester. Between 1972 and 1974, Mott had taught mathematics and computers at Oberlin College in Ohio, then had moved to Boston with the intention of working for a year before applying to business school.

"I went to the Boston office of Scientific Data Systems looking for a job in computer sales support," says Mott. "I was a long-haired freaky looking kid at the time, and they were skeptical. But they knew about Darwin's project and suggested I go talk to Ginn. I talked to Darwin, and a week later he invited me back to meet some of the people from PARC. Two weeks after that, I was in Palo Alto."

Mott's introduction to PARC was humbling. "I was totally at sea technically. There were so many new things going on that I had never seen before. And no documentation existed to help me teach myself. I felt I had been thrown in at the deep end."

After a month of intense self-education, Mott reached a conclusion that neither he nor Darwin Newton had anticipated—POLOS was far too complicated and incomplete to support the word processing and printing applications Newton wanted to try at Ginn.

The complexities of POLOS arose from its unique heritage, one dating back to 1945. That year, Vannevar Bush, who coordinated all federally funded scientific research during World War II, published an article entitled "As We May Think" in *The Atlantic Monthly*. In it, Bush warned that by relying on antiquated methods for storing, researching, and transmitting information, the post-war world risked losing control over the accumulated record of its own knowledge and history.

"The summation of human experience," he claimed, "is being expanded at a prodigious rate, and the means we use for threading through the consequent maze to the momentarily important item is the same as was used in the days of the square-rigged ships."

Bush speculated on a series of hypothetical information management technologies, the most intriguing of which he called "memex." "A memex," he wrote, "is a device in which an individual stores his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory."

Memex users would manipulate keyboards, screens, micro-

film, and photographic technology to build "trails" of pertinent information from any knowledge base, no matter how extensive or fast changing. As a result, Bush predicted, "wholly new forms of encyclopedias will appear, ready-made with a mesh of associative trails running through them, ready to be dropped into the memex and there amplified. The lawyer has at his touch the associated opinions and decisions of his whole experience. . . . The physician, puzzled by a patient's reactions, strikes the trail established in studying an earlier similar case, and runs rapidly through analogous case histories, with side references to the classics for the pertinent anatomy and histology. The chemist, struggling with the synthesis of an organic compound, has all the chemical literature before him."

When Douglas C. Engelbart, then a naval radar technician stationed in the Philippines, read Bush's *Atlantic Monthly* piece in 1945, he reacted to it with the same dedication of purpose fired in Bob Taylor many years later by J.C.R. Licklider's article "Man-Computer Symbiosis." Following the war, Engelbart earned college and graduate degrees in electrical engineering, and immediately began pushing his academic and industrial colleagues for the chance to develop the information tools Bush had envisioned. But throughout the 1950s, most people considered Engelbart's proposals mystical. Memex-style computing required interactivity; interactivity implied direct access to computers; direct access, in the days of batch processing, seemed out of the question for both technical and economic reasons.

Only the advent of timesharing made Engelbart's dream a possibility. He landed a job at the Stanford Research Institute and, while there, gained the support first of Bob Taylor at NASA, then Licklider and Taylor at ARPA for a laboratory dedicated to building interactive hardware and software on a timesharing computer base. He called the effort the "Augmented Human Intellect Research Center," although it became better known as the Augmentation Research Center, or ARC.

Engelbart sought to augment human intelligence through ordering the information on which people rely. His philosophy and leadership inspired ARC to invent hardware and software dominated by rules of hierarchy and structure. In the paper accompanying their famous 1968 demonstration of interactive computing tools, coauthors Engelbart and Bill English explained that ARC had "adopted some years ago the convention of organizing all information into explicit hierarchical structures, with provisions for arbitrary cross-referencing among the elements of the hierarchy.

"3c2b1 The principal manifestation of this hierarchical structure is the breaking up of text into arbitrary segments called 'statements,' each of which bears a number showing its serial location in the text and its 'level' in an 'outline' of the text."

The conference paper itself was presented as an example of Engelbart's text structuring. Every paragraph (including the one just quoted) started with an alphanumeric label in the manner of the classic outlining system learned by most high school students. Engelbart and English reported that, when made dynamic by a computer processor, their design facilitated a number of varied and well ordered perspectives on information. For example, users could check logic and completeness by requesting the screen to display "statements" having the same outline level of importance; they could quickly locate specific supporting material by selecting the appropriate alphanumeric heading; they could expand and tie together topics by creating a subheading at the most appropriate point in a given body of thought.

The impact on work and productivity at the Augmentation Research Center was pervasive:

> 3c4 The basic validity of the structured-text approach has been well established by our subsequent experience.
>
> 3c4a We have found that in both off-line and on-line computer aids, the conception, stipulation, and execution of significant manipulations are made much easier by the structuring conventions.
>
> 3c4b Also, in working on line at a CRT [cathode ray tube] console, not only is manipulation made much easier and more powerful by the structure, but a user's ability to get about very quickly within his data, and to have special "views" of it generated to suit his need, are significantly aided by the structure.
>
> 3c4c We have come to write all of our documentation, notes, reports, and proposals according to these conventions. . . . We have found it to be fairly universal that after an initial period of negative reaction in reading explicitly structured material, one comes to prefer it to material printed in the normal form.

Engelbart called the collection of hardware and software "NLS" for "oN-Line System." To computer scientists in the ARPA com-

munity, NLS represented the most impressive advance of interactivity since the invention of timesharing, and it inspired many subsequent developments, including much of the work at PARC.

Engelbart, however, was apparently a querulous boss. In 1970, Bill English led a group of researchers from the Augmentation Research Center in Menlo Park, California, to the new Xerox PARC facility in Palo Alto. They brought with them two of Engelbart's deepest biases: faith in structuring and reliance on timesharing. Both contributed to the failure of the POLOS office system project.

Unlike the decision of the Computer Science Laboratory to abandon timesharing, English's group in SSL sought to elaborate, even to multiply, it. They designed POLOS around a dozen separate timeshared computers, with each computer dedicated to a specific office function. For example, one computer would handle filing, one editing, one printing, and so on. Users were to sit at terminals with simultaneous access to all computers. It was an unbelievably complex scheme.

"English's group," says Tesler, "selected POLOS, in part, for conservative reasons. They didn't think personal computers would be economically viable. So they decided to build what they called a 'distributed system.' As it turned out, their approach was conservative economically but too aggressive technically."

The toughest job in a timesharing system is building the operating system, the software that enables the central processor to direct traffic. In both batch processing and personal computers, the operating system enables the central processor's control unit to keep order among input, output, memory, arithmetic, and logic. To this, timesharing adds the burden of avoiding collisions among multiple users. For example, preventing User A's input from disrupting User B's program or User C's output.

POLOS elevated this degree of difficulty by yet another order of magnitude. Its operating system would have to manage the logistics of *different computer operations* (input, output, memory, arithmetic, and logic), *different users* (Secretary A, Executive B, Clerk C), and *different office functions* (editing, filing, printing).

POLOS condemned its users to the same lack of predictability that plagued all timesharing systems. Furthermore, it continued to emphasize structured text editing. English's team was confident that memos, letters, proposals, and other documents coming out of a POLOS office would reflect a higher quality of logic and thought because of the system's outlining tools. All of which was fine in theory and acceptable to computer scientists comfortable with advanced systems. But it made no sense for Ginn because, as Tim Mott realized, most people do not comfortably compose in a hierarchically restricted fashion.

"Ginn needed fairly simple programs for word processing and page layout," says Mott. "The POLOS application was modeled after NLS which in turn was based on the Bush paper. They were really tools for organizing thought, not for editing and page design. While the word processing functions were there, there was also lots of other stuff that Ginn would not have needed and were too cumbersome. I didn't think the editors at Ginn would take the time required to learn. My model for this was a lady in her late fifties who had been in publishing all her life and still used a Royal typewriter.

"Also the POLOS hardware was too complicated to be moved out of the research lab. The hardware was wrong, the operating system and approach were wrong, and the applications were wrong. I was bummed out when I realized I had to write a letter to Darwin, almost a letter of resignation, saying 'I finally have a handle on what's going on here, and, believe me, it's not going to do you any good.' "

To Mott's relief, Tesler suggested they consider employing the Alto-Ethernet-EARS-Bravo system at Ginn. One of the first versions of Bravo had just been completed by Lampson, Simonyi, and a team from CSL. While this early edition of the program did not include the formatting and typeface capabilities needed for page layout, it did contain the basic word processing features sought by Darwin Newton for his editors. With English's permission, Tesler and Mott dropped POLOS and asked Lampson and Simonyi if they would agree to a series of modifications to make Bravo easier to learn and operate.

Tesler considered Bravo terrific with one exception—it reflected the habit among computer scientists of requiring users to memorize special instructions to enter what are called "modes" in order to manipulate the program. A mode is a means of putting the machine into a certain context to get it to execute a desired task. The shift key on a typewriter is a mode. Unless it is depressed, the typist cannot produce capital letters or any of the uppercase symbols represented on the number keys.

In Bravo, editing changes required a "command" mode. First,

the operator used his mouse to select a letter, word, or section of text to be edited. Then by pressing the "D" key, the Alto would delete the selection. Similarly, in the command mode, "I" would permit insertions of text, "R" would replace text, "U" would undo the effect of the previous command, "G" would get a document from memory, and "P" would put it back. If "D," "I," "R," "U," "G," or "P" were typed outside the command mode, they would merely appear as letters in the user's text.

Computer scientists like modes because they support flexibility in programs by allowing a variety of keystroke combinations to transmit a range of electronic meanings to the processor. For novices, however, modes can be treacherous. One leading commentator on computer anxiety writes, "There is a story, probably apocryphal, told about another text-editing program that has separate input and command modes. According to this story, a hapless user wanted to type the word 'edit' into his document. Unfortunately, the program was in command mode when he started typing—the 'e' selected everything currently in the document; the 'd' deleted everything that was selected; the 'i' caused the program to enter insert mode; and the 't' inserted the letter "t." The result: the entire document was replaced by the letter 't.' Sorry."

Mott agreed with Tesler that Ginn employees who, to a person, had little understanding and big fears of computing, were more likely to welcome word processing that was modeless. With the cooperation of Lampson and Simonyi, the two men went to work. For two months, they worked every day at their Alto in overlapping fourteen hour shifts to rewrite Bravo's code. They named their modeless program "Gypsy" after the costume worn by Mott's stepdaughter that Halloween.

When they finished, PARC had the first computer editing program easy enough to be learned in a few hours by people who had never touched a computer. As Tesler explains, "In Gypsy, there were no modes. 'I' meant 'I' and nothing else. If you wanted to insert text, all you had to do was position the mouse at the point in the text you wanted to change, click the button, and begin to type."

By February of 1975, Mott was back in Massachusetts installing the system for its initial test. "When it came time to begin instructing people about Gypsy," Mott remembers, "I went straight for the lady with the Royal typewriter, figuring if I could teach her, it would be clear sailing for the rest. After a few hours of coaching, she had learned enough to go off and use the system on her own. A few days later, she said the quality of her work had improved because she was always dealing with clean copy, and it was easy to make changes. She volunteered that she couldn't imagine having ever worked any differently."

The experiment at Ginn was a huge success. Darwin Newton estimated the system would save between 15 and 20 percent of the cost of editing Ginn books. While some editors balked at the new technology—"I'd rather be home with a tall drink than be at the office with a machine," one reported—most agreed with the assessments of Newton and the woman with whom Mott worked first. Indeed, the enthusiastic Ginn employees even proposed that PARC add more features to the program, including a computerized thesaurus.

Larry Tesler received a letter of commendation from Xerox and a large raise; Tim Mott, after spending the rest of 1975 and part of 1976 coordinating the introduction of the system at Ginn, was hired to work full time at PARC. They were deservedly proud of their contribution.

"Gypsy," Tesler points out, "was the first thing outsiders in the company understood. Ginn loved it. It was so easy to learn. They would hire temporary typists to come in at eight, they were trained on the system by nine, and typing full speed by ten. This was revolutionary in publishing. In a couple of years, they were using this word processing for over fifty percent of their books. Nowadays, so what? People use one hundred percent word processing. But this was 1975."

People throughout PARC shared Tesler's excitement. The creative effort behind Alto, Ethernet, EARS, Bravo, and Gypsy also had produced advances in programming languages, operating systems, drawing and painting programs, and communications. Of course some ideas, like POLOS, did not pan out. But as most researchers at PARC happily acknowledged, the POLOS project simply confirmed the willingness of Xerox management, and especially Goldman and Pake, to underwrite some failure as an inevitable cost of successful research.

Furthermore, the Ginn experiment coincided with a number of schemes in 1975 and 1976 that sought to transfer PARC's technology out of the laboratory. A Xerox division in Southern California requested permission to make and market a laser printer, a patent was applied for and received on the Ethernet, and headquarters approved the formation of the Systems Development Division (SDD) to engineer PARC inventions into products. By mid-1975, SDD had signed up a number of PARC's most illustrious stars, including Ron Rider, the designer of the Research Character Generator, Charles Simonyi (Bravo), Robert Metcalfe (Ethernet), and Chuck Thacker (Alto).

That spring, another CSL engineer, John Ellenby, got permission to redesign the Alto and set up a production line to manufacture the computers in small quantities instead of one at a time. A year later, Ellenby's project had gone so well that he was encouraged to submit a proposal on the Alto to a Xerox task force then deciding on new product strategies for the company. And by August of 1976, rumors were spreading at PARC that the task force had selected the Alto as Xerox's entry into the word processing market just then emerging in the United States.

Xerox's investment, inventiveness, and effort seemed ready to pay off with an "architecture of information" every bit as powerful and provocative as the one Peter McColough had predicted in his speech at the beginning of the decade. Thinking back to the accomplishments of PARC's first five years, Chuck Thacker says, "It was certainly from my own experience the largest continuous piece of creative output that I have seen anywhere. And it was like being right there at the Creation. A lot of people worked harder than I had ever seen, or have seen since, doing a thing that they all thought was worthwhile, and really thought would change the world."

"In general, the broader issues of integrating all the functions in an office and recognizing the kinds of procedures and systematic ways of behaving that the customers already have in their offices and selling equipment that will help the customer improve that kind of work flow is not very well understood—either by our sales force or by many of us developers."

Massaro, however, confidently expected to mold a team capable of selling the Star. But Xerox denied him the opportunity. In 1982, a year after Dallas introduced the Star, David Kearns decided to integrate Massaro's sales force into the larger copier sales organization. Explanations for the move varied—jealousy among the copier salespeople because Dallas's computer sales force called on higher-ranking customer executives; a desire to present "one face" to the customer; a recent consolidation of sales forces by Kearns's alma mater, IBM; the continuing failure of Massaro's division to turn a profit.

Ironically, the decision was buttressed in part by the success of another Massaro scheme. In late 1981, he had persuaded Kearns to ask the copier sales force to push Xerox's new electronic typewriter, called the "Memorywriter," and by spring, the product was already a hit. Selling the Memorywriter, however, required practically no adjustment to the experience and training of the copier salespeople. "The Memorywriter's a simple product," said Jack Crowley. "An ape could sell that. It's not a system."

Massaro, who had been a model for Kearn's decision to reorganize the copier group around strategic business units headed by their own general managers, now saw the Office Products Division stripped of its valuable sales function. It was too much. He told Kearns he was quitting. "I didn't want to lose my salesmen," he explained later. "That was basically the reason why I left Xerox, the fact that they were combining sales organizations."

Many people in Stamford cheered; they had never liked Massaro, and in their still too political world, Kearns's persistent praise for the Dallas executive made them despise Massaro even more. The news disheartened Massaro's supporters. At his farewell party in Dallas, they presented him with a bronze statue of the Road Runner. Said Massaro, "These were three of the best years of my life. In the first two, I was a hero; in the third, I fell on my sword, and so I was an asshole."

Chapter 20

David Liddle resigned soon after Don Massaro, and together they founded their own company to pursue systems applications for personal distributed computers. They were not, however, either the first or the last employees to leave Xerox because of disappointed computer expectations. By 1983, more than a dozen key contributors had taken their knowledge and experience elsewhere. Some former PARC scientists, like Alan Kay, Tim Mott, and Charles Simonyi, assumed research or product development posts in established firms; others, including John Ellenby, Patrick Baudelaire, Chuck Geschke, and Robert Metcalfe, raised venture capital to start new enterprises. Meanwhile, with each departure, Xerox lost a measure of control over the ideas that made its "architecture of information" so special.

At times, Xerox was bizarrely generous with its computer inventions. For example, in late 1979 one of the company's investment arms contacted Steven Jobs of Apple Corporation about a possible deal. Jobs, who for years had heard about the fabled accomplishments of Xerox PARC, asked for and received a tour of the research center. According to Larry Tesler, who conducted a demonstration of the Alto for Jobs, the young entrepreneur immediately grasped what had eluded Xerox executives for more than half a decade.

" 'Why isn't Xerox marketing this?' " Tesler recalls Jobs demanding. " 'You could blow everybody away!' "

Ensuing discussions between Xerox and Apple fizzled. But within months of Job's visit, Tesler left Xerox for Apple, and Jobs ordered an Apple team to design the "Lisa," a computer introduced in 1983. The Lisa replicated many features invented at Xerox, and because of Apple's strong presence in the personal computing market, the Lisa seemed to steal a march on Xerox's Star.

"Office equipment analysts have started referring to PARC-style systems as 'Lisa-like,' not 'Star-like,' " noted a reporter. "Apple's next computer, MacIntosh, scheduled to ripen into a commercial product by the end of this year, could further identify

Apple with PARC's ideas. The engineering manager for MacIntosh came from PARC, where his last big project was a personal computer."

The Jobs boner upset a number of PARC people who believed the Xerox investment group should have been looking for ways to invest in developing, not disclosing, PARC's ideas.

"I was just as incensed at them for not taking advantage of the things that were here as I had been with Bob Potter for ignoring them," said George Pake. "To allow Jobs to see the power of the system and gain access to bright people was a dumb thing to do. And he did make off with Tesler. Once he saw it, the damage was done; he just had to know that it was doable. Just like the Russians and the A-bomb. They developed it very quickly once they knew it was doable."

The defections of both people and ideas caused a lot of embarrassment for Xerox; Pake's friends teased him about PARC being a "national resource." In September of 1983, *Fortune* ran a major article entitled, "The Lab That Ran Away From Xerox." Featuring Ellenby, Simonyi, and Tesler, the piece criticized Xerox both for letting so many talented researchers escape and for failing to profit from the computerized office systems invented at PARC.

In one respect, though, the *Fortune* account missed its mark—the most productive of Xerox's computer scientists remained at PARC. Bob Taylor, Butler Lampson, Chuck Thacker, and others who had created personal distributed computing still preferred research to commercial opportunity. Since the mid-1970s, they had designed a new processor, called the "Dorado," which was at least ten times as powerful as the Alto, then had employed it to develop a variety of pathbreaking operating systems, programming languages, and other software, all of which had perpetuated PARC's distinction as among the very best computer science establishments in the world.

Nevertheless, PARC had drifted from the tranquil intellectual retreat Jack Goldman and George Pake had founded thirteen years earlier. In 1983, tensions threatened to break the center apart. And at the root of the unhappiness—like so many of PARC's attainments—was Bob Taylor's obsession with interactive computing.

From the beginning, contradictions had beset Taylor's position at PARC. George Pake had coveted Taylor's access to the finest young computer minds in the country, but not imagining that

such talent would expect and depend upon Taylor's guidance, had withheld the Computer Science Lab manager's job from the former ARPA official, citing Taylor's inadequate personal research track record. Rather than confronting Pake with the misunderstanding, Taylor had contrived the artificial "Mr. Outside/Mr. Inside" plan that had led to Jerry Elkind's appointment as manager of the Computer Science Lab. Although Taylor and Elkind had forged a modus vivendi, the two men could not shield the strains in their relationship from others at PARC.

"Jerry came thinking it was a real job," said one PARC administrator. "He started behaving like someone who was going to manage CSL, and that immediately put him in conflict with Bob."

Despite Elkind's title, the crucial organizational attributes of the Computer Science Lab—collaborative hiring, flat structure, constant communications, and habitual use of the systems invented—each bore Taylor's stamp. Scientists cleared their budgets, project plans, and other decisions through Elkind, but most of them discussed the issues at length with Taylor first. The reality of their respective positions frustrated both men.

"Eventually," said Ed McCreight, a researcher, "I came to notice that Elkind was not giving Taylor much rope, and I didn't think Bob was enjoying it, although Taylor always seemed to keep things about himself to himself."

Apparently, Elkind's personality further diminished his authority within the lab. In contrast to Taylor's contemplative and solicitous manner among the researchers, Elkind tended to attack people and their ideas. Although some scientists thrived on his belligerence, many more were disheartened, driving them further into Taylor's camp.

"Elkind had an awful shorthand," comments McCreight. "He would say, 'Explain why what you're doing is important!' in a way that implied he didn't think it was important. I came from a small town in Pennsylvania where that kind of tone bordered on an attack. So I would think that maybe my stuff really was dreck. Later I learned in my head not to interpret it so negatively. But my gut reaction always overwhelmed my mind. I was really getting down on myself."

Agrees Severo Ornstein, who worked for Elkind at Bolt, Beranek and Newman as well as at PARC, "Jerry just didn't know how to encourage people."

In mid-1976, Elkind took a leave of absence to serve on Bob

Sparacino's technical staff for a year. The awkwardness of serving two bosses dissolved. So many researchers appreciated the improved atmosphere that, just prior to Elkind's return in late 1977, a handful of them petitioned George Pake to arrange for a different assignment for Elkind.

"I had lunch with George," recalls Ornstein, "and told him that the lab just operated better under Bob than Jerry, that Bob was easier to work with. When I said that, George looked off into the distance, his eyes glazed over, and he said, 'Taylor can be pretty hard too.'

"Then George said he had promised Jerry his job would be held for him at the time he took the leave and that he wouldn't go back on his word. So a number of the senior people in CSL took our courage in both hands and went to Jerry and told him we'd rather he didn't come back, that we liked it better with Bob in charge."

In the words of one participant at that meeting, the scene was "surrealistic"—Elkind, who in effect had been hired by his subordinates in the Computer Science Lab was now informed by the same group that they no longer wished to report to him. All agreed that Elkind handled the situation well. Although he did not resign, he left the Computer Science Lab a few months later to manage Xerox's new Advanced Systems Division. Thus, by early 1978, Bob Taylor finally gained in name what had been his in fact for seven years: the job of managing the Computer Science Laboratory at PARC.

George Pake found the Elkind coup attempt distasteful; according to Jim Mitchell, Pake unfairly charged Taylor with engineering the event. Though unsubstantiated, Pake's suspicions could not have surprised anybody at PARC. Almost from the beginning, he had considered Taylor a zealot who fostered an "orthodoxy" within the Computer Science Laboratory.

"CSL under Taylor," charges Pake, "was not hospitable to differing points of view. You could get shot down so fast if you didn't agree with the party view."

Taylor's single-minded pursuit of interactive computing threatened the most basic premise of Pake, the university man—truth's dependence on diversity of opinion. Yet those who defend Taylor claim that Pake's standard for research, formed by decades as an academic physicist, misconstrued the nature of computer science. Unlike physicists, computer scientists don't assemble tools

to verify hypotheses about the world. Rather, they build software and hardware because they believe, in Chuck Thacker's words, such constructs "will be a good thing to have." Specific engineering goals, not truth, governed the Computer Science Laboratory; where Pake feared orthodoxy, Taylor's partisans saw only effective management.

"The fundamental ethos in a university," comments Butler Lampson, "is that the professor is an independent agent. A department or a school derives its strength from individual achievements and not from the fact that they work together. The thing that's good about it is that anybody who has a new idea can pursue it without being affected by the group.

"We didn't run CSL that way. We promoted a substantial amount of group thinking. What that means is that we didn't hire people whose thinking was too far afield from the main line of the group.

"The reason we ran CSL that way was that it was our view that if you were going to do systems research, you had to get people to work together to a considerable extent or else you couldn't get the kind of synergy that was necessary. In order to do that, you had to have good planning, because if everyone did their own thing, then there would be no way to get those things to work together. That was why we enforced that. There was a corresponding danger. There was a hazard. But we didn't miss too many good things, too many good people as a result."

Without question, CSL achieved the goals they set for themselves. Most of the researchers in the Computer Science Lab unanimously credit Taylor for their mutual success. "I've never worked for anybody better in my life," said one; "He's been in many ways the best manager I've ever seen, let alone worked for," agreed another. Still, Taylor's monomania and the "group thinking" Lampson described could and did ostracize some CSL scientists who, for whatever reason, sought independence.

"Being inside CSL and being on Taylor's good list was like heaven," says Ed McCreight. "You could do anything, and really conquer the world. Bob really cared about you as an individual, and if you had any problem with Xerox, Bob would take care of it. You felt entirely supported.

"But it was bimodal. If you weren't on his good list, it must have been like hell."

Some disgruntlement stemmed from the inevitable hierarchy that crept into CSL's organization notwithstanding Taylor's preference for a "flat" structure. As the age, tenure, talents, and friendships of several of the researchers matured, people like Lampson, Thacker, Mitchell, McCreight, and Ornstein emerged as the opinion makers to whom Taylor looked for technical guidance. The group even had a name, the "Graybeards," and their influence rattled some of the lab's younger members who felt they didn't get enough time and attention from the boss.

Other dissatisfied researchers lost out in labwide decisions over technical direction; a few simply disliked Taylor or Lampson, the lab's major technical presence. As one malcontent is reported to have complained, "I don't want to be told all the details of my personal life by Bob Taylor, and I don't want to be told all the details of my professional life by Butler Lampson."

All the outcasts found their way into George Pake's office, confirming Pake's fears about the dangers of orthodoxy. Equally unsettling to Pake, Taylor appeared set on spreading the CSL view of PARC's objective to the research center as a whole: he placed "agents" like Alan Kay in the Systems Science Laboratory; he criticized the series of men who managed SSL; he belittled the accomplishments of the physicists in the General Science Laboratory; he questioned Pake's judgment and understanding of the computer research conducted at PARC. The Bob Taylor seen by George Pake in the PARC-wide context differed from the calm, understanding, and supportive manager of creative egos resident in the Computer Science Lab. Instead of the image of a reassuring ear, the metaphor Pake and other PARC managers prefer is that of a fanatic, vituperative mouth. To them, Taylor would stop at nothing to get his way.

"One of the things that makes Taylor so powerful," says Frank Squires, a longtime advisor to Pake, "is that he can focus so strongly on narrow goals and never be distracted. He never has to worry if he's right because he never has to consider anything beyond his narrow focus. One aspect of this was to take to attacking other activities outside CSL. He managed CSL by circling the wagons; everyone inside is great, and everyone outside is a turkey. It led to a whole series of actions—some above the table and some below—that were critical to other groups. And that was the beginning of the end of his relationship with Pake."

Taylor scoffs at such complaints. He contends that the PARC achievements most consistent with Peter McColough's original "architecture of information" charter were produced by those scientists who, whether officially part of CSL, SSL, or other labs, cooperated within the vision of personal distributed computing. Throughout the 1970s and early 1980s, Taylor laments, those people constituted less than a fourth of the center's budget, an allocation of resources that he claims precluded Xerox from enjoying even greater accomplishments. In his opinion, Pake's evenhanded university ethic veiled an unacceptable ignorance—both about the nature of computer science and the best interests of Xerox.

"I came to Xerox," asserts Taylor, "driven by a vision and set of objectives, and having decided to devote my life to work that was going to make a difference. Pake is motivated differently than I. I'm a content guy. Pake's more of a process guy. From my perception, he simply wasn't interested in the content of what we were doing.

"We were exploring a new space: personal distributed computing. That space was huge, in terms of the number of unknowns that were associated with it, the number of problems one ought to have worked on if you were really to have explored that space. We never had the resources to explore the way we should have. And the reason that we didn't is that because from our management on up there wasn't a sufficient understanding of what the nature of that problem space was.

"When we talked to Goldman or Pake about the importance of programming technology, for example, they would nod their heads. But as soon as any issue would come up where we wanted to increase our *investment* in programming technology, in one way or another, the answer would always come back, 'Not at the expense of physics!' "

Taylor conceived of PARC solely in terms of interactive computing; he rejected as misguided any activity not in furtherance of his goal. Consequently, he persistently criticized Pake's support for the research being conducted by other PARC laboratories into the physics of optics and materials. All major breakthroughs in interactive computing after the invention of the integrated circuit, Taylor argued, had depended, and would continue to depend, on solving the mysteries of design and programming logic instead of the physics of light, electricity, or matter.

"The complexities that we had to work out to do what we did were *not*," he stressed, "in physical science."

Unfortunately, Taylor packaged his profound differences with

Pake in a series of wild charges and unrealistic demands. Every time Pake refused to sacrifice physics research in favor of Taylor's activity, the screams about underfunding would fly. To Pake, Taylor's relentless complaint about resources must have seemed like a wolf on the wrong side of the door. That such operating groups as Scientific Data Systems and the engineering division led by O'Neill and Sparacino would grab for research money was, perhaps, predictable; but that one research manager would so blatantly demand the budgets of his peers deeply offended Pake.

Nonetheless, as long as Pake directed PARC, Taylor's stridency was kept in check. In 1978, however, Pake unexpectedly relinquished day-to-day control of the West Coast research center. That spring, to protect the research function's corporate reporting status from an assault by Sparacino and O'Neill, Pake had agreed to replace Goldman as the company's chief of research. Bob Spinrad, a man with a computer engineering background, assumed Pake's position at the head of PARC. The switch pleased Taylor and troubled his enemies in equal measure.

"Spinrad and Taylor got along," according to Bert Sutherland, the manager of the Systems Science Lab, "because Taylor found in Spinrad a computer guy who shared his world view. The word around PARC became, 'When Spinrad needs to know what to do, he asks Taylor.' "

A year after Spinrad took control of PARC, Xerox headquarters asked each of the research centers to provide a long-term assessment of direction and related budgetary requirements. Taylor contends that Spinrad diligently reviewed the technical priorities of each of PARC's laboratories, including the strengths and weaknesses of their respective scientists, before recommending any action.

"This was a job that had never been done before," claims Taylor. "Pake certainly never did it. After about a year and a half of reviews, Spinrad started a series of final meetings designed to build a five-year plan for PARC. The plan that emerged would have gradually changed PARC's investment profile over the period of the plan, enlarging the computer investment at the expense of the physics investment."

The physicists immediately complained to their fellow physicist Pake, who was already unhappy with Spinrad for a separate reason. Shortly after Pake had assumed Goldman's corporate responsibilities, he had persuaded Kearns—over Sparacino's bitter

objections—to add an integrated circuit laboratory to PARC. Despite repeated admonishments from Pake, however, Spinrad had failed to hire a manager for the new lab. Now Spinrad's five year plan, no doubt under Taylor's spell, devalued other research efforts Pake considered important. Therefore, to get action on the integrated circuit lab and to protect his vision of PARC, Xerox's head of research intervened.

"I was unhappy with Spinrad's report," explains Pake. "Not only the substance, but also the way it was handled. It was no way to inspire creativity to tell people they had no future. In some of the labs to be deemphasized there were good people and good projects making progress, all of which I believed Xerox should keep.

"There was also the second problem. We had the approval for the integrated circuit lab, and we had to get a first-rate manager. Bob Spinrad had been interviewing without success, and, to his credit, admitted as much in his performance review that year.

"So I took the artificial step of splitting PARC into two. One part, called the 'Systems Center,' reported to Bob Spinrad, and the other, the 'Science Center,' to Harold Hall."

Pake put the new integrated circuit lab, the threatened physics lab, and certain computer science groups that were out of favor with Taylor into Harold Hall's Science Center, intentionally restricting Spinrad's control to those activities supported in his long range plan. Pake admits that his action bore an obvious message for Taylor.

"There was a method to my madness. I wanted a plurality of viewpoint and did not want everything done here within what I called the orthodoxy of CSL. There is no question it was a brilliant lab. But it did not have to be so dominant."

Buildings, however, do not divide as easily as organizations; the scientists of Hall's Science Center and Spinrad's Systems Center continued to park in the same lots, walk the same halls, and eat in the same cafeteria. But their mutual enmity rapidly spoiled the common atmosphere. According to Bill Spencer, the man Hall hired away from Bell Labs to run the integrated circuit lab, Xerox's West Coast research "Centers" suffered from a social pathology —people went out of their way to physically avoid and verbally assault their antagonists. Taylor supporters accused Pake of double dealing; Pake's allies labeled Taylor a malevolent "Reverend Moon" who had mesmerized researchers within and beyond CSL to accept his word as gospel. Moreover, they claimed Taylor had begun an effort to "sell" his entire research group to other companies.

Spencer, the new integrated circuit lab manager, got to know Taylor quite well. The two men played tennis nearly every Saturday throughout 1981 and 1982. "After every tennis session," says Spencer, "we would have drinks—Taylor a Dr. Pepper and me a beer. He was a man of very deep habits, and very strong feelings. Most of my views about PARC were very much influenced by Taylor. Taylor would say over and over again, 'We've pulled together a group of outstanding people, and we've demonstrated we can build things. The whole world says it's great stuff, yet Xerox hasn't made a success of it. We now need more funds, and if we could only get rid of other parts of PARC, we could go on and do more great things.' "

Based on his twenty years of experience at Bell Labs, Spencer concluded that Taylor's group had an inflated opinion of themselves. Nevertheless, he appeared to win Taylor's friendship without succumbing to Taylor's orthodoxy, a feat that persuaded Pake to tap Spencer to put PARC back together again in the spring of 1983.

"There was an artificiality about the two 'Centers,' " explains Pake. "And I believed Spencer was strong enough to think through the future of PARC in ways that would not arbitrarily curtail any segment of it. Taylor had always been a big managerial problem, and I thought it was important that they be able to get along. I thought Spencer was the ideal man to run the lab because he and Bob played tennis together and got along great."

Spencer avoided Bob Spinrad's error. Rather than buying into the Taylor line, PARC's new director turned his full attention to an issue he considered far more critical to the research center's long-term viability. He made clear that he believed Xerox and PARC had failed to join scientific invention to commercial innovation and that at least half the blame belonged to the research center. To improve results, he asked PARC's lab managers to get to know their "customers" elsewhere in the corporation. "Fifty percent of their performance reviews," Spencer decreed, "would be based on how successfully they achieved it. What I was looking for was to see if you could couple PARC closer to the organization and still keep it as creative as it had been in the past."

Spencer's independence quickly cooled his relationship with Taylor. The two men stopped their game of tennis and, ominously, quit talking. Reflecting bitterness over what happened, Spencer says, "It wasn't my tennis elbow. Bob feels threatened by technical people who want to move into management. He's an interesting guy to have come as far as he has without the academic trappings people usually have in those positions. It would have been hard for him to have been an individual contributor."

From Taylor's perspective, George Pake appeared to have cloned himself—right down to Spencer's retreat to credentialism. Taylor was wrong. Significantly, Bill Spencer lacked the depth of Pake's conviction about pluralism. For years, those who had objected to Taylor's ethics and behavior had advised Pake to fire the CSL strong man. To have acquiesced, however, would have struck at the core of Pake's principles. Although Pake may have considered Taylor obnoxious, as long as CSL's "orthodoxy" was kept at bay, Taylor had an equal right to pursue research in George Pake's organization.

Pake and Taylor coexisted for more than a decade; Spencer, as PARC director, confronted Taylor within months. In late August of 1983, Spencer summoned Taylor to his office, briefly delineated their differences, then handed Taylor a memorandum of their "understanding." In it, Spencer accused and forbade Taylor from attempting to induce employees to leave the company, gave him three weeks to reorganize CSL into several subgroups, directed Taylor to share managerial responsibilities with others in CSL, commanded CSL to improve its contacts with the rest of Xerox, announced that Spencer himself would attend CSL staff and planning meetings, and ordered Taylor to stop bad-mouthing other PARC programs and labs.

The Spencer note ended with an ultimatum: "Bob, it is my desire that as the result of our discussions that you will make the necessary corrections in your behavior and actions to be a valued member of my staff. However, it is important for you to understand that any failure to comply with these action items, or the confidentiality of this memo, will result in disciplinary action that may include your termination."

Spencer threatened Taylor with much more than a lost job. To re-create CSL's computing environment, even with the same people, would take several years because, while many researchers might join a fired Taylor elsewhere, their technology—blueprints, programs, and manuals for the personal distributed computing hardware and software advances of the past decade—would remain at Xerox. They could pack their ideas and experiences, but to reach the same interactive computing frontier in a different setting, a Taylor-led laboratory would have to retake much of the ground already conquered. No one as creative as Taylor or his colleagues likes to reinvent old wheels.

Nevertheless, Taylor refused to back down. A week after Spencer had read him the riot act, CSL's manager submitted a lengthy defense of his conduct at PARC. He wrote for the record that he had spent his entire Xerox career recruiting talented people to join, not leave, PARC, that his organization worked effectively without subunits, and that senior people in CSL either preferred research to management or rejected managerial opportunities because of "the management environment immediately over me." Taylor was dismayed by Spencer's charges, but remained, true to form, unbowed:

"The PARC research investment strategy," he argued in his memo to Spencer, "has not been one of building upon our strengths to maximize synergy, but rather to let many disjointed flowers bloom. This investment strategy is most unwise for computing research in particular.

"The important issue here and implied throughout your entire memo, is the successful transfer of technology within Xerox, is it not? This is what we all want! Indeed, that is why all the people in CSL joined CSL. We recruited to the dream that Xerox, using the technology we believed we could create, would significantly change and enhance the information world for millions of Xerox customers. We intended from the beginning to put the technological flesh on the bare bones of Peter McColough's phrase, 'the architecture of information.' That is what we came here to do, and we have made a very good beginning; certainly better than any other group in the world, large or small, over the past thirteen years. This is especially remarkable when one realizes that we have worked with less than one-fifth of the total resources of PARC from the beginning through to the present.

"We have transferred an enormous amount of technology. But the real challenge has been the transfer of an entirely new and quite different framework for thinking about, designing, and using information systems. *This is immensely more difficult than transferring technology*. Opportunities for pioneering completely new ways of thinking about large collections of ideas are rare. Most people spend a lifetime without any such opportunities. Over the

past twenty years, I have been fortunate to have been a leader in three: timesharing; long-distance, interactive networking; and personal distributed computing. Each of these required large upheavals in the way people think about information systems. I learned a great deal from these experiences."

Though Taylor knew his PARC days were numbered, he insisted that his note to Spencer was not a letter of resignation. Before taking that final step, Taylor wanted the chance to discuss his position with his senior colleagues—the "Graybeards"—as well as David Kearns. Spencer okayed a Taylor-Kearns meeting, but officiously refused to lift his ban against disclosure of the situation to anyone at PARC. After failing to win any comfort from Kearns, Taylor informed the Graybeards that he would resign.

On September 19, two weeks after publication of "The Lab That Ran Away From Xerox," the curtain rose for the final part of the drama missed by the article in *Fortune*. Taylor convened the Computer Science Lab for the last time. In a brief address to an audience that included Bill Spencer, he reviewed the circumstances of the previous weeks, announced his resignation, and, before leaving, thanked the researchers for enriching both his life and the world with their many accomplishments. The scientists were stunned.

According to Jim Mitchell, "Spencer stayed on after Taylor departed and tried to be upbeat. But no sooner did Spencer begin talking than Chuck Thacker stood up and publicly resigned. That stopped Spencer cold. He recovered to tell Thacker that he knew things were emotional, that nothing was irreversible, and why didn't they talk the next day, et cetera. But the next day, Thacker was gone."

The other Graybeards went directly to Pake to tell him they thought a terrible mistake had been made. They warned him that Xerox might lose much of the lab unless Taylor was reinstalled. Pake explained that he had no desire to see CSL dissolved, but that, in Spencer's opinion, Taylor had become intractable, creating a crisis in the management of the center. To the Graybeards, it sounded as though Pake disclaimed any part in the decision to fire Taylor. They weren't convinced.

"Spencer may have been the hit man," one of them remarked, "but I know who the godfather was."

Over the next few months, the group tried without success to persuade Pake, Spencer—even David Kearns—to invite Taylor back. Meanwhile, Taylor received an offer from Digital Equipment Corporation to build a computer research center in Palo Alto. Word about Taylor's deal spread through PARC with the same effect that the news of Taylor's move from Utah to PARC had had on the scientists of Berkeley Computer Corporation more than a decade before. Lampson immediately resigned to join Taylor and Thacker at DEC, and more than a dozen top researchers followed. When they arrived, they found a relieved Bob Taylor.

"It's great," he exclaimed, "to finally work for a computer company!"

For George Pake, the crisis had come just when PARC seemed to have earned a more active role in the corporation. In 1983, the PARC inspired laser printer business continued to grow rapidly, Xerox headquarters approved plans to back product developments in the high density memory disk and solid state laser technology pioneered at PARC, and, perhaps most important, PARC's Ethernet had been modified and installed in Xerox's successful "10 Series" of third generation copiers.

"Xerox researchers can be very proud," Pake wrote, "that there is no current or projected major Xerox product which is not enabled in some technologically essential way by a result from the research laboratories."

As for the Alto and the Star, the optimistic Pake refused to mourn the personal computer opportunity that had bypassed Xerox. Showing that he remained more physicist than businessman, Pake asserted that, if anything, Xerox had introduced personal distributed computing in the Star too soon.

"The public," he claimed, "wasn't ready."

Epilogue