Virtual Machines, Virtual Infrastructures:
The New Historiography of Information Technology

Published in *Isis,* 1998

Review of:

Campbell-Kelly, Martin; Aspray, William. *Computer: A History of the Information Machine*. x + 342 pp., illus., bibl., index. New York: Basic Books, 1996. $28.

Cortada, James W. *Information Technology as Business History: Issues in the History and Management of Computers.* (Contributions in Economics and Economic History, 177.) xiv + 263 pp., index. Westport, Conn./London: Greenwood Press, 1996. $69.50.

Norberg, Arthur L.; O'Neill, Judy E. (with contributions by Kerry J. Freedman), *Transforming Computer Technology: Information Processing for the Pentagon, 1962-1986*. With contributions by Kerry J. Freedman. (Johns Hopkins Studies in the History of Technology, 18.) xvi + 360 pp., illus., tables, index. Baltimore/London: Johns Hopkins University Press, 1996. $49.95.

Hafner, Katie; Lyon, Matthew. *Where Wizards Stay Up Late: The Origins of the Internet.* 304 pp., illus., index. New York: Simon and Schuster, 1996. $24.

Rochlin, Gene I. *Trapped in the Net: The Unanticipated Consequences of Computerization*. xvi + 293 pp., index. Princeton, NJ: Princeton University Press, 1997. $29.95.

Landauer, Thomas K. *The Trouble with Computers: Usefulness, Usability, and Productivity*. xiii + 425 pp., illus., tables, bibl., index. Cambridge, MA: MIT Press, 1995. $15.

Do computers really "compute"?

Everything a computer does reduces to binary arithmetic. Yet at the level that matters to users — the level of word processors, Web browsers, or weather forecast simulations — computation is only sometimes the significant feature. Software programs, together with the socio-technical systems in which they are embedded, turn the physical "computer" into an infinite variety of *virtual machines.* This is the meaning of the ugly but useful term "applications." As Thomas Landauer puts it, "it is a mistake to view the computer as a single technology. More appropriately, each major application is a new technology harnessing information processing capability, much as the electric motor, the locomotive, and the jet plane all harnessed energy-transforming capability" (p. 104).

It has taken historians a long time to understand this point. The main paradigm in computer historiography has been device history, focusing on machines, engineers, and the computer industry. Ironically, the history of software and applications remains the field's most underdeveloped area, by far. To varying degrees, the six books under review represent a new perspective on computer history: computers in use, or what we might call the history of virtual machines. The computers-in-use perspective marks a mature historiography that seeks a many-leveled, situated, and ultimately critical view of computers in sociocultural context.

In *Information Technology as Business History,* James Cortada argues that business applications of computers are the key not only to computer industry development patterns, but also to the computer/society intersection more generally. Cortada is a historian's historian, a master of sources and archives and the author of many books on computer history, including three indispensable bibliographic guides. Though trained in history, he spent over 25 years at IBM in sales and marketing. This collection of previously published essays covers three main topics: the structure of the computer industry, computer applications, and corporate management and use of information technology.

Cortada believes that particular applications make their way into general use only when they become marketable. In the device-history paradigm, marketability is usually tied to absolute cost, cost/performance ratios, and hardware innovation (especially in memory size, processor speed, and reliability). These factors certainly determined the limits of affordability and performance. In the mainframe era, which lasted until about 1965, only very large organizations could afford computers at all, and only those with major computational needs — in the literal sense — could justify their expense. By the minicomputer era of the 1970s, medium-sized businesses could afford them. With the advent of personal computers (PCs) around 1980, computers became a consumer and small-business product.

But as Cortada demonstrates, hardware and cost were only part of the story. Proactive marketing techniques developed by business machine companies in the late 19th century persisted into the computer era. In what Bruno Latour might call an "enrollment" process, vendors expended considerable effort to "convert" their customers to the new machines. They then provided massive technical support, often including extensive formal education. Widespread service networks guaranteed customers rapid, on-site technical aid, but also provided vendors with intensive feedback. This vendor-driven information loop was the primary factor in the success of digital computers and of the companies that sold them.

Cortada's excellent work deserves wide influence. Unfortunately, his editor has served him poorly, passing over significant defects of presentation and style. Several grammatical errors appear on almost every page, and the book careens haphazardly between a chatty clarity and dry, highly technical discussions accessible only to professionals.

In *Computer*, Martin Campbell-Kelly and William Aspray describe a related phenomenon of the PC era (which Cortada tends to treat as an afterthought.) Amateur hobbyists — not the business-oriented computer industry — first created PCs. Although they borrowed off-the-shelf hardware, the mid-1970s idea of a private, microprocessor-based computer belonged uniquely to them. So did the crucial ultra-compact PC software; mainstream industry simply couldn't be bothered. Perhaps misled by the device-history paradigm, Campbell-Kelly and Aspray make the odd mistake of defining what counts as a "personal" computer by size, when it should be defined functionally. In the early 1960s, Wes Clark and others marketed several computers designed for a single user. *Computer* ignores these precursors. Despite such minor failings, this is the best comprehensive history of computers now available. It makes an excellent textbook.

The authors argue that many PC industry success stories owed less to the individual geniuses lionized by the popular press than to sheer luck and high-powered consumer marketing — the PC industry's largest single expense. Apple, for example, engaged the prestigious Regis McKenna public relations firm. IBM spent many millions on its "Little Tramp" campaign, appropriating the *Modern Times* image of Charlie Chaplin as workingman's hero. Understanding PCs as consumer products explains some apparently bizarre phenomena, such as Apple's hiring of former Pepsi-Cola executive John Sculley as its CEO. With PC mass markets, media hype and the packaged software industry largely replaced the customer education campaigns of hardware vendors.

These descriptions of marketing's key part in creating social roles for computers tie in with another of Cortada's arguments. Unlike most device-history accounts, Cortada's image of computer development is evolutionary, not revolutionary. He argues that the business machine industry always understood the new machines as logical successors to existing equipment.

Industry leaders allowed military and scientific researchers to develop the field in 1945-55 not because they did not understand it, but because the expense of electronic digital techniques was still prohibitive. As soon as cost-performance ratios began to decline, they jumped in.[1]

Computer adopts a more measured variant of this perspective. Its authors confirm Cortada's argument that the modern computer's major precursors were office machines: punch-card tabulating equipment, typewriters, and cash registers. Indeed, IBM, NCR, and other office machine companies, many with 19th-century roots, became the eventual leaders in electronic computing. Unlike Cortada, however, Campbell-Kelly and Aspray see the 1945-55 decade as one of genuine uncertainty, when few saw much commercial potential for the unreliable and ultra-expensive electronic computer. Well into the 1950s, electromechanical equipment could perform most business tasks much more cheaply and practically.

What *uses* of computers finally justified their expense to a critical mass of corporate customers? Cortada identifies three main periods. An *initial implementation* phase (1952-1965) saw adoption by sectors with heavy computational needs: banking, accounting, insurance, manufacturing, and inventory control. In the second phase, *major adoption* (1965-1981), other uses emerged: retail, where key innovations included point-of-sale technology and the Uniform Product Code; on-line office communications and accounting; word processing; and "decision support" systems. Finally, *extensive implementation*, beginning in 1981, saw computers applied to every area of business. Major new applications during this period included spreadsheets, office networking, desktop publishing, and mobile computing.

Campbell-Kelly and Aspray also point to crucial new techniques of the key period 1955-1975. But they emphasize fundamental underlying capabilities, such as real-time computing and time-sharing, rather than applications. These techniques bridged the gap between expensive scientific computing and business uses. Real-time systems, for example, enabled computers to handle high-volume, time-critical operations such as airline reservations. Time-sharing made expensive machines more cost-effective by permitting many people to use them at once.

Where did these techniques come from? Neither Cortada nor Campbell-Kelly and Aspray pay quite enough attention to the crucial role of military support. In 1945-55, roughly 75 percent of

---

[1] Others have recently reached similar conclusions. Pugh, Emerson W., Building IBM: Shaping an Industry and Its Technology (Cambridge, MA: MIT Press,

1995).

all computer R&D funding in the U.S. came either directly or indirectly from the military, which wanted the machines for nuclear weapons research, aerospace engineering, and code-breaking. Without the Cold War, which drove military budgets sky-high and oriented a whole society toward high-technology armed forces, computers would doubtless have evolved more slowly.[2] The SABRE real-time airline reservation system, for example, was a direct offshoot of the gargantuan SAGE computerized air defense system.

Significant government support did not end when the computer industry matured in the mid-1960s. Arthur Norberg and Judy O'Neill drive home this point in *Transforming Computer Technology*, their authoritative study of the Information Processing Techniques Office (IPTO) of the Defense Department's Advanced Research Projects Agency (ARPA). Since its founding in 1962, IPTO has probably done more than any other single institution (possibly excepting MIT and IBM) to guide the long-term course of computer development.

Most of *Transforming Computer Technology* chronicles IPTO-funded projects in its four most important research areas: time-sharing, computer graphics, networks, and artificial intelligence. Notably, all of these areas are software-driven. They are not applications *per se*, but fundamental computing techniques. Time-sharing, graphics, and networks form the core of the modern computing paradigm; in this broad sense, IPTO may be directly responsible for computers as we know them today.

During the period covered by this book, IPTO was much beloved within academic computer science. Its reputation derived directly from an unusually informal management style. Instead of peer review, proposals were evaluated directly by IPTO's tiny staff, always drawn from the highest levels of the field. Thus IPTO "viewed need from the perspective of... the computing community itself" (p. 15). The agency focused on building large "centers of excellence" at a few institutions, such as MIT and Stanford, with minimal bureaucratic overhead. IPTO typically supported blue-sky projects whose military and/or commercial payoffs might not materialize for a decade or more.

IPTO's beneficiaries have virtually nothing negative to say about the agency. Indeed, one problem with *Transforming Computer Technology* is that the authors have not thought to investigate how those *excluded* by the centers-of-excellence strategy viewed IPTO. Virtually all those interviewed for the book were ARPA administrators or grant recipients, and it often reads like a *Festschrift.* The book never really comes to grips with IPTO's second-order impacts (such as effects on Silicon Valley and Route 128), which may have been far larger than its moderate budgets would suggest.

However, the authors do have much to say about one crucial issue: IPTO's role in creating new military technology. Most of its beneficiaries have seen ARPA as a sort of alternative NSF, interested mainly in quality research, not booty for its military paymasters. But *Transforming Computer Technology* shows quite clearly that this was true only in the agency's earliest years, if ever. Advancing computers for military "command and control" was IPTO's fundamental goal from the very beginning, and IPTO managers worked constantly to translate research into useful military products.

The most famous IPTO product was the ARPANET computer network, developed in the late 1960s. Well covered by Norberg and O'Neill, the ARPANET is also the subject of *Where Wizards Stay Up Late*, by journalists Katie Hafner and Mark Lyon. Did IPTO build the ARPANET to help ARPA-sponsored researchers share information, or to test techniques for

---

[2] Edwards, Paul N., The Closed World: Computers and the Politics of Discourse in Cold War America (Cambridge, MA: MIT Press, 1996).

"survivable" military communications? These books don't settle that controversial question, largely because neither follows the story beyond the research community into the Pentagon itself. (*Wizards* handles its interviewees even more credulously than *Transforming Computer Technology*.) But Norberg and O'Neill demonstrate that within a very few years of the first ARPANET hook-ups, ARPA directors were promoting similar packet-switched networks for military command-control-communications systems. The ARPANET itself served primarily in testbeds and research programs, but networks based on ARPANET technology rapidly found their way into active military systems.

Wizards explores the human dimension of ARPANET development. Two-thirds of the book tells the story of the first few years, when researchers at Bolt, Beranek, and Newman developed the Interface Message Processor (IMP), a small, dedicated computer for linking mainframes to the ARPANET. ARPANET site operators had only to program their computers to communicate with the IMP, a relatively straightforward task. Hafner and Lyon labor mightily to make this highly technical story interesting, but they don't always succeed. Most of the excitement derives from endless all-nighters endured in races to meet various deadlines. By contrast, the last third of the book is fascinating. Here they recount the spontaneous emergence of uses for, and communities around, the network.

The most important use was, and remains, electronic mail. Email had existed on local time-sharing systems long before the ARPANET. Around 1972, ARPANET users began to write programs to facilitate email. These spread like wildfire. By 1973, a study "found that three-quarters of all traffic on the ARPANET was email" (p. 194). Another intriguing ARPANET story regards the anarchical but amazingly efficient hacker communities that designed the all-important protocols which govern the structure and handling of network messages.

The ARPANET grew rapidly throughout the 1970s, but most users were research communities in government and universities. Meanwhile, businesses were building their own private, local area networks. Many of these used Ethernet technology, developed at the Xerox Palo Alto Research Center (PARC) in a group led by former IPTO director Robert Taylor. (The PARC group also developed the first commercially marketed graphical user interfaces.) By the mid-1980s, many desktop computer systems came with built-in networking capabilities.

The need to connect many independent networks led to the concept of an internet, or network of networks (of which *the* Internet is simply the most important). None of the books reviewed here fully cover this recent story. *Wizards* ends in the mid-1980s, when the ARPANET was gradually phased out. Soon afterward, the NSF sponsored the Internet's transition to private management, and commercial entities began to discover its value. Internet "takeoff" occurred in the early 1990s, largely because of the Internet-based World Wide Web hypermedia access system. An especially important point, hardly mentioned in these accounts, is the major role of Internet cost structures (based on connection speed, *not* traffic volume) in facilitating the network's explosive growth.

Although not primarily a work of history, Gene Rochlin's *Trapped in the Net* addresses the historical significance of computer networks perhaps better than any book to date. (Caveat: I refereed this book, and an excerpt of my report appears on its back cover.) Rochlin notes that beginning in the late 1970s, PCs spread through businesses in an extremely haphazard way. First individuals, and later departments, typically purchased their own machines. The vast variety of systems created chaotic, extremely inefficient conditions for exchanging information. This led to the phenomenon of "sneaker net" (hand-carrying floppy disks), and caused the emergence of a very large meta-level industry in compatibility products. As their companies computerized underneath them, upper-level managers often lost overall control of information processing. Rochlin sees office networking in the second half of the 1980s, and Internet linkage in the 1990s, as a management-led effort to reassert centralized control. Cortada confirms these points, with quantification.

Trapped in the Net also analyzes other computer-related social changes. For example, Rochlin argues that the network-based globalization of financial markets is rapidly eroding the power of government regulatory agencies to stabilize and police them. Such agencies traditionally function through leverage on a few central nodes, such as the major stock exchanges. As networks move trading into decentralized, global domains beyond the reach of national governments, this leverage is fast diminishing. The book explores similar phenomena in manufacturing, air traffic control, and military systems (including an especially good chapter on the Persian Gulf War).

Rochlin's book contains a crucial policy message. Computers make it easy to create huge, highly integrated systems, and also to alter their structure at will; I call these *virtual infrastructures*. As their size and level of integration increases, these can fail in highly complex, far-reaching, and dangerous ways. Rochlin predicts catastrophic outcomes if we do not soon find better ways to address these infrastructural levels of responsibility and control.

The Trouble with Computers, whose sociotechnical systems perspective resembles Rochlin's, may be the most important book about computers to appear in this decade. One reason is that Landauer zeroes in on the question most important to commercial users: payoff.

Landauer tries to sort out the causes of the so-called "productivity paradox" in computerization. "Productivity," in economics, is a measure of workers' output per hour; the economic boom of the industrial era is generally credited to the productivity improvements provided by mechanical automation. The "paradox" is that despite over $4 trillion spent on computing in the United States since 1960 — and current annual expenditures of about 10 percent of GNP — the rate of productivity growth in the U.S. economy *declined* in the early 1970s and has never rebounded to its 1950s highs. In the sophisticated financial sector, where computerization began in the mid-1950s, a dollar invested in information technology in 1973-83 produced just one dollar of output — a zero return on investment. If computers are really the revolution that will end the mechanical age, why has this colossal outlay failed to pay off?

Economists have been worrying about this problem for well over a decade. Its causes are complex, and information technology inefficiencies are only one. Nevertheless, a survey of the economics literature convinces Landauer that the paradox is real, despite the "excuses" offered by a few optimists to explain it away. He then draws the bold and necessary conclusion: computers don't work very well. They are hard to use, unreliable, and often applied where they offer few benefits. Worst, businesses rarely incorporate computers into a comprehensive *socio*-technical design process. Landauer analyzes a fantastic array of all-too-familiar situations where computers fail, confuse, and generally wreak havoc upon systems they are supposed to improve. There is a salutary bubble-bursting quality to Landauer's brilliant discussion of examples; reading this book will certainly darken your view of computer "power."

Landauer divides the history of computers-in-use into two main phases: an automation phase (roughly 1960-1972) and a "decision support" phase (1973-present). Here Landauer locates one explanation for the productivity puzzle. Decision support — computer assistance in organizing and processing information for people to use — has proven extremely complex. Attempts directly to automate information-handling jobs generally don't work. Instead, these jobs need to be approached as problems in the "organization of organizations." Another issue is the *over*use of information technology: too many reports, too much detail. Finally, like Rochlin, Landauer notes the nightmarish integration problems created by the tangled web of incompatible legacy systems.

Landauer wants to cut this Gordian knot with “user-centered design,” an iterative cycle of testing and redesign already popular in Europe. Since the earliest days of computing, programmers have generally tested software on themselves. But as Landauer demonstrates, their extreme expertise makes programmers *least* able to find and resolve the problems that plague ordinary users.

This brings us back to the point with which I began. Only a small minority of those who use computers in their work actually use them *as computers*, i.e. as general-purpose information machines. Instead, most users — checkout clerks, telemarketers, air traffic controllers, etc. — encounter only one or a couple of virtual machines (applications). Many, perhaps most, computer-based systems actually *prevent* users from programming them. Landauer argues that only testing in actual use — studying real users’ behavior and desires, evaluating new systems against control groups, and using the results to force even radical redesign — can ever yield the productivity improvements computers promise. Unfortunately, user-centered design may itself prove to be a pipe dream; today’s ultra-rapid product cycles will make it almost impossible to deploy.

The historiography of computers-in-use represented by the excellent works reviewed here reaches the social significance of computers in a way the device-history paradigm never can. Yet in order fully to capture the historical importance of computerization, we will need an even sharper focus on the integrating role of computers in sociotechnical systems. Let me suggest how this might be accomplished.

The modern computing paradigm integrates at least six major functions. In approximate order of historical appearance, these are calculation, simulation, control, information processing, communication, and visualization. We are now witnessing a further integration, rapidly engulfing all pre-existing media, known as “digital convergence.” The path from giant electronic calculators to this embedded, networked, global, all-media infrastructure was anything but obvious in the 1940s. Far more than hardware, software, and applications, the story involves the building, bridging, and reconstruction of a myriad of human institutions.

A history of digital convergence and virtual infrastructures might reveal a very large iceberg, of which *individual* computers and even widely adopted applications are only the visible tip. I am thinking, for example, of the effects of computerization on telecommunications of all types; of microprocessor control of everything from cars to cruise missiles; of the new scientific understandings made possible by computerized visualization of gigantic data sets; and of the amplified globalization of financial markets and commerce brought about by computer-based networks. It is at these infrastructural levels, where disparate functions were gradually integrated into what is fast becoming a global, infinitely reconfigurable infrastructure, that the computer is achieving its original promise as the “universal machine.”