History and Technology, 1998, Vol. 15, pp. 7-29
Reprints available directly from the Publisher
Photocopying permitted by license only

Y2K: MILLENNIAL REFLECTIONS ON COMPUTERS AS INFRASTRUCTURE

PAUL N. EDWARDS

Program in Science, Technology & Society, Bldg. 370 Rm. 111, Stanford University, Stanford, CA 94305-2120

Abstract: Computers have become the control, information storage, and information processing technology of choice in many other, pre-existing infrastructures. This essay argues that historians of computers and information technology should expand their agenda to include the origins and impacts of this phenomenon. Studying computer-based infrastructures could lead to a new historiographical approach focussing on 'internetworks'. These are very large, integrated, extremely heterogeneous metasystems, made possible in part by 'digital convergence' or the ability to record, store, process, and distribute information in all media using computers and computer networks. Key actors include the developers of protocols for information exchange among heterogeneous networks.

Ten years ago, Michael Mahoney published an important essay on "The History of Computing in the History of Technology."[1] Now, near the eve of the millennium and in the forum of this special issue of *History & Technology*, seems a propitious moment to reexamine the state of the field.

I do not propose to provide here a comprehensive review of computer historiography. Instead, I want to reflect on the emergence of global computer-based infrastructures. I believe these point to the need for a new historiography, one that so far has not often been attempted, perhaps because its scale and scope are so daunting. Still, I think we are ready at least to ask how an understanding of computers as infrastructure might point the way to new approaches in the history of computers.

THE MILLENNIUM BUG

At five minutes to midnight on December 31, 1999, computer system operators around the world will not be at parties swilling champagne. Instead, they will be perspiring in front of terminals, praying to whatever they hold most dear that their Millennium Bug software patches will work. Many of their prayers will not be answered.

The Millennium Bug – also known as the Year 2000 Problem, or Y2K, in computerese[2] – is a forty-year-old software time bomb. On its face, the problem seems trivial. To conserve computer memory (at the time, a relatively scarce resource), the first generations of computer programmers typically used date fields consisting of six decimal digits (MM/DD/YY). The year prefix "19" was simply assumed. This entirely common practice continued well into the 1980s.

At the turn of the millennium, software incorporating this assumption will act as if the year is 1900. The results are unpredictable. Spectacular breakdowns in everything from military command-control systems to Internal Revenue Service tax computations are highly likely. Microprocessors embedded in everything from automobiles to elevators may fail. Rumor has it that some airlines are not accepting flight reservations for January 1, 2000, for fear of major confusion in the worldwide air traffic control system. Articles on the crisis have appeared in the *New York Times*, *Newsweek*, and other leading general-audience publications. Strangely, although we know exactly *when* this technological breakdown will occur, we cannot predict exactly *what* will happen.[3]

The Y2K problem has, in fact, already arrived. Credit card companies and banks are already wrestling with the problem of credit and debit cards whose expiration dates are "00" or later.[4] There have been frequent reports of Y2K-related failures in credit-card verification devices at retail stores. Visa recently threatened to impose fines of up to £100,000 on British banks that fail to make their software "Y2K-compliant," as industry jargon puts it.[5] The satellite-based Global Positioning System (GPS) contains ultra-accurate atomic clocks that are used to calibrate time by other systems around the world. Perhaps the Y2K problem can be solved for the GPS itself. But that will not be enough. Some one million pieces of end-user equipment reliant on GPS store dates in a thirteen-bit format which will overflow around August 20, 1999. No one knows exactly what the result will be. Since the computers of many large financial institutions rely upon GPS time calibration, one good guess is disturbances in international financial markets, where very-short-term interest-bearing transactions are commonplace.[6]

Programmers are frantically working to eliminate Y2K problems. Even so, reliable estimates project that fewer than half of all large software systems (those containing over 500,000 lines of code) will be fixed in time to avoid problems. World total cost projections for fixing the problem are many billions of dollars; one estimate – probably high, but from a reputable source – is $70 billion. The author of this estimate notes that these costs represent perhaps "the largest and most expensive technical problem in all of human history."[7]

Why is this bug so expensive to find and fix? The worst of many problems hindering programmers in this effort is "legacy software." In many older institutions, software written in the 1960s, or even before, is still in use. The computer languages in which these programs were originally written have changed, most of them drastically. Some are no longer in use. The original programmers are long gone. The code they left behind is often undocumented, its original purpose and function now difficult or impossible to divine. Much legacy software no longer exists as source code (i.e., in the high-level language in which it was written). Instead, only object code (compiled machine-language programs) remains. Decompiling this object code is a major headache, when it is possible at all. For these reasons and many others, Y2K is likely the thorniest generic problem ever faced by computer programmers.

Y2K resembles many other technological problems resulting from what Edward Tenner has called "the revenge of unintended consequences."[8] Examples of such problems are hardly difficult to find. The U.S. interstate highway system linked cities, but also hastened the decline of passenger railways and helped create suburban sprawl and endemic air pollution. High-tech agriculture based on hybrid seeds, pesticides, machinery, and artificial fertilizers raised crop yields worldwide, but also created human health problems, disrupted ecosystems, and burdened the Third World with debt and dependency. Antibiotics saved millions of lives, but also caused the evolution of super-bacteria that resist every known drug. In each case, technological solutions were so widely adopted that they became fundamental sociotechnical systems. Their very success eventually caused other, equally severe problems – most of them never imagined by the system-builders who promoted the original solutions.[9]

Similarly, the Y2K problem exists because of precedents set in the 1950s and 1960s for what were, at the time, very good reasons.

Few programmers ever even considered that their software might still be in use after 1999. Fewer still thought of themselves as designing a global infrastructure. Yet many of their programs not only survived, but went on to become the core ever-more-elaborate systems accreted, like tree rings, over decades. Future generations of programmers continued to use the MM/DD/YY and other limited date formats simply because those were the standards set by their predecessors. Only in the 1990s did the realization spread that what was once a feature had now become a bug.

The Millennium Bug will affect a substantial fraction – estimated at between 5 and 15 percent – of *all* installed software. While newer software, such as that written for personal computers, is less likely to be affected, it is far from immune. Thus Y2K will disrupt virtually all applications of computers, across all sectors, to an as-yet-unknown but certainly significant extent. Therefore, it is most appropriately conceptualized as an infrastructure problem.

COMPUTERS AS INFRASTRUCTURE

The term "infrastructure" seems to have a military origin, referring to fixed facilities such as air bases (OED). Contemporary usage has grown much broader. The *American Heritage Dictionary* defines "infrastructure" as (1) "an underlying base or foundation, especially for an organization or a system," and (2) "the basic facilities, services, and installations needed for the functioning of a community or society, such as transportation and communications systems, water and power lines, and public institutions including schools, post offices, and prisons." In the latter sense, the term is perhaps best defined negatively, as those systems without which contemporary societies cannot function.

We have heard a lot of talk about computer infrastructures in the last few years. Al Gore promoted a "National Information Infrastructure" during the 1992 presidential election campaign; now there is even a "Global Information Infrastructure."[10] These concepts refer mainly to the burgeoning Internet and World Wide Web. The popular "information superhighway" analogy indicates a general propensity to think of this infrastructure as a *physical* system, perhaps including fiber optic transmission lines or a satellite relay network.

But the Y2K example forcefully demonstrates that computer networks are not the only way in which computers have become infrastructural in the developed world. Even more important, computers have become the control, information storage, and information processing technology of choice *in many other, pre-existing infrastructures.* Indeed, it is difficult to imagine how such fundamental systems as telecommunications, electric power networks, banking, stock markets, air traffic control, or government could function if all computers were suddenly to vanish from the face of the earth.

Computers have become, as it were, the infrastructure of our infrastructures. As such, they exhibit properties different from those of more traditional infrastructures, which might be characterized as large technological *systems.* (Good examples of the latter are electric power networks, municipal water supplies, and railroads.) Computer-based infrastructures include systems, but they also operate at the meta-level we have learned to call *internetworks*. I will say more about this below.

What could historians learn from an approach to computers as an infrastructural technology? I believe this will be an important next step in the historiography of information technology, and perhaps even of technology in general. In the rest of this essay I explore these implications and sketch some examples.

FROM HARDWARE HISTORY TO SOCIOTECHNICAL SYSTEMS

Mahoney's essay pointed out that the historiography of computers has progressed through a series of genres. As the first generation of digital computer pioneers matured and then retired, "insider" histories appeared: memoirs, retrospectives, biographies, and in-house corporate histories. These studies focused mostly on early computers, their ancestors, and the intellectual and technological problems surrounding the construction of logic machines from digital hardware. Typically, they treated the design histories of landmark machines, such as the ENIAC and EDVAC. A second genre, the interview-based journalistic account, has been popular since digital computing's earliest years. Finally, what Mahoney called "social impact statements" have emanated from sociologists, media specialists, and computer professionals.

Only recently, Mahoney noted (in 1988), had studies by professional historians begun to appear. Like their predecessors – now doing double duty as primary sources – these were marked by an overwhelming emphasis on hardware and engineering. In addition, because early computers were mostly conceived and used as giant calculators, early professional histories emphasized their mathematical and scientific uses.

As I have pointed out elsewhere, this emphasis on hardware and calculation created a historiographical paradigm with certain unfortunate consequences.[11] The real historical importance of computers lies not in their calculating power alone, but in their ability to integrate previously unrelated, highly heterogeneous functions within a single technological framework. The most important of these, in approximate order of historical appearance, are calculation, simulation, control, information processing, communication, and visualization. Together, these functions form the core of the modern computing paradigm, i.e. networked, distributed computing with graphical user interfaces. Even more important, they allow computers to link other kinds of devices into large, integrated metasystems – and thus to create new infrastructures.

Device history is fundamental. Nevertheless, it tends to obscure the origins and impacts of this larger trend, whose importance in the second half of the 20th century easily compares with the world-changing role of internal combustion engines and their vast associated infrastructure (oil wells, gas refineries, distribution networks, paved roads for automobiles, shipping, air transportation, diesel trains, etc.) in the first half. Before returning to this point, let me briefly review some recent historiographical trends upon which a history of computers as infrastructure might be built.

The general-purpose digital computer, as a physical machine, is useless without the software that transforms it into a myriad of special-purpose *virtual machines*. Hardware, being visible and expensive, is merely the easiest and most obvious aspect of "the computer" to study. Little has yet been written on software, an ultimately more significant aspect of computing. Most extant studies focus on the history of computer languages and programming methods.[12] Programming, although important, is really a meta-level from the perspective of computers in actual use. Focusing exclusively on programming is like trying to grasp the historical meaning of internal combustion engines by studying the machine tools used to build them.

New efforts in this area are proceeding apace, however. For example, James Cortada has energetically argued that a business-history perspective can generate a more applications-oriented approach to computing. On this view, computers are evolutionary offshoots of older business technologies, such as punch-card tabulating equipment and typewriters, rather than the revolutionary tools of science portrayed by many journalists and computer pioneers. Business history focuses attention on how computers – initially expensive, unreliable, and extremely difficult to use – were made into salable commodities. This effort, of course, involved not only technical improvements, but also concerted marketing efforts and "customer education" campaigns. Such a focus necessarily leads to a study of applications, the ultimate selling point of any business machine.[13]

The "social informatics" perspective represents another, parallel line of research, stemming mainly from sociology and anthropology.[14] It explores the relationships between information technology, organizations, and social change. Unlike Mahoney's "social impact statements" – which consisted mainly in general, frequently ungrounded praises and laments of computing's social effects[15] – social informatics emphasizes meticulous empirical studies of the entire context of computers in actual use. In the early 1980s Rob Kling, a pioneer of social informatics, drew attention to the "web of computing," i.e. the complex interaction between organizational structures and technological change.[16] Unsurprisingly for historians of technology, this is almost never the one-way street imagined by eager marketeers. This field's very sophisticated scholarship now includes a whole battery of case studies of how organizations *actually* absorb and employ new computer technology.[17]

More recently, scholars have begun to focus on the important role of government and the military as funders, purchasers, and promoters of computer technology. The well-known stories of such government-backed infrastructure projects as the SAGE air defense system[18] and the ARPANET and Internet[19] have been filled out by a growing number of English language studies of government's role in countries other than the United States, including the Soviet Union and its satellites.[20] Brazil, for example, promoted an indigenous computer industry by means of import barriers and lavish R&D funding.[21] Especially in the cases of the US and France (which had a national plan for computer development from the 1960s, and built the first national computer network, the Minitel, in the early 1980s), these government-led efforts have frequently shaped computer-based infrastructures and paced their introduction.

Some scholars have adopted a Chandlerian[22] approach that explores the origins and impacts of information technology on a global, decades-to-centuries scale. This difficult method's signal advantage is its necessarily functionalist approach, which focuses attention on the application rather than the technical content of the new technology. Its drawback is a pronounced tendency toward technological determinism.[23]

Perhaps the best example of both the virtues and the problems of this approach is James Beniger's quirky, flawed, but still profoundly important book *The Control Revolution*, aptly subtitled *Technological and Economic Origins of the Information Society*.[24] A sociologist by training, Beniger tends to build arguments from lists of "firsts" and anecdotal sketches, rather than the kind of fully contextualized, deeply researched accounts most respected by historians. Worse, he overgeneralizes, stretching the concept of control so broadly as to encompass nearly everything, from the computer to the origins of life itself. Yet he also develops an absolutely crucial insight regarding the critical role of information, and therefore of information technology, within every aspect of modern capitalist economies.

Beniger sees information technology innovations – not only in calculating devices, but also in communication, control, and organizational (bureaucratic) technology – as part and parcel of the Industrial Revolution. In the 19th century, he argues, high-volume factory production rapidly saturated local markets. In order to sell their surfeit of mass-produced goods, manufacturers were forced to develop extensive distribution networks. As the speed and geographical reach of the industrial extraction–production–distribution system grew, the latter required increasing speed, breadth, and sophistication from the information and communications networks that connected points of production with points of sale. These networks constitute the feedback control systems of the industrial era. Thus computers, for Beniger, are simply the latest technology in a long line of innovations driven by the need to manage (control) the production, distribution, and sale networks of increasingly far-flung capitalist enterprise, all driven, ultimately, by the unstoppable engine of mass production.

Similarly, Manuel Castells' monumental three-volume study *The Information Age: Economy, Society and Culture* begins with a historical examination of the global "informational economy."[25] The *Information*

Age series, a masterwork whose significance can hardly be overstated, builds on Castells' lifetime of research and teaching on five continents. *The Rise of the Network Society*, its first volume, explores the role of computers and telecommunications in a new "informational mode of development."

Castells distinguishes his concept of modes of development, or "the technological arrangements through which labor acts upon matter to generate a product," from the traditional Marxian concept of modes of production, or "the social relationships of the production process, for example, class relations ... [or] capitalism."[26] In the informational mode of development, information itself is both a raw material and a product. This feature generates an ever-faster development cycle; since each new process or product consists largely of information, it can instantly become input to a new round of innovation.[27] Information technology thus plays a double, and doubly important, role as the fundamental basis not only of information products and processes, but of the global organization of *material* production and distribution as well. The informational mode of development takes different forms in different world regions, with material production concentrated in some areas and information production focused elsewhere. But information technology, he argues, creates everywhere a "networking logic" that integrates specific technologies into larger systems.

All of these approaches converge, in different ways, on a historiography of computers as elements of large sociotechnical systems à la Thomas Parke Hughes.[28]

Hughes described the latter as driven by the visions of *system builders* such as Thomas Edison or Alexander Graham Bell. As Hughes puts it, Edison wanted to build not a light bulb, but a lighting system that would encompass electric power generation and delivery as well as lighting. System builders often worked with *heterogeneous components*, including organizational and social as well as technical elements. Key events in the history of large systems include the resolution of *reverse salients*, or stubborn socio-technical problems that hinder the realization of the overarching system. Typically, large sociotechnical systems are centrally controlled, with one or a few major functions. Examples include the railroad, telephone, and nuclear early warning systems. Finally, Hughes signaled the importance of *technological momentum*, a kind of critical-mass effect which makes fundamental change in a system difficult to achieve once it has been widely adopted.

Hughes's work has led historians of technology to attempt the daunting task of large-systems historiography.[29] Despite the enormous difficulty of this work, the results have been gratifying. Large-systems histories have refocused the field on subjects more obviously of current value than traditional concerns with particular technical objects and their inventors. They have forced historians to reconsider and recontextualize the very concept of "technology." They have brought with them collaborations and contacts with other disciplines concerned with the social role of technology, such as sociology and cultural anthropology. Finally, and perhaps most importantly, they have contributed to a rapprochement between the history of technology and "mainstream" historical studies.[30]

What if we were to rethink computer historiography in terms of infrastructure, in the same way that Hughes rethought the historiography of technology in terms of systems?

HISTORY OF COMPUTERS AS HISTORY OF INFRASTRUCTURE

Consider the business buzz-phrase "digital convergence." It's an apt description of the massive, rapid shift to digital formats now underway in *all* communications media, including text, television, telephone, video, audio, and photography.[31] Digital convergence means that each of these can now be recorded, stored, processed, and distributed by computers and computer networks. This process is not only technical, but also commercial, social, and political. It opens the door to integrated infrastructures of huge scale and scope.

For example, Microsoft's recent moves into broadcast television, global satellite systems, and Internet service are designed to acquire a controlling position in a possible integrated network of digital television, Internet, and telephone service. Similar things can be said of financial systems at all scales, from on-line home banking to global stock markets, and of many, perhaps most, other socio-technical systems upon which developed societies are based. In this way the enormous latent scope of the phrase "information technology" – too often taken merely as a synonym for computing in a narrower sense – is presently being made manifest.

I do not think we yet have anything like a satisfactory account of how this ever-accelerating convergence came about. How and why did computers journey from their initial applications as calculators and text processors in business, government, science, and the military to their current role as general-purpose tools for all communications media? How did digital formats come to be (or seem) better, cheaper, or more flexible than the older analog alternatives?[32] How did such an enormous variety of devices and social processes come to count as "information technology"? Who were the important system builders, and what were the key reverse salients? Is the language of systems adequate to describe these phenomena, or do they require a new, or an additional, set of concepts?

The historiography of large technological systems has demonstrated how difficult it is even to conceptualize – let alone successfully to accomplish – an approach of the sort I am proposing here. As Thomas Misa has observed, the historian must chart a course between the Scylla of a grandiose technological determinism, à la Chandler, and the Charybdis of an over-detailed, impossibly dense exposition which loses track of the overall goal in an attempt to respect the complexity of the subject matter.[33] The scales involved are so vast, across the manifold geographies of time, space, technologies, organizations, and cultures, that few single individuals can ever expect to master it. Perhaps this is a subject best tackled by teams – an uncommon work structure for historians, but one that seems increasingly necessary in the historiography of large-scale technological phenomena.

NETWORKS OF SYSTEMS

An even worse problem stems from the very nature of computer-based infrastructures. Some (not all) of these are essentially different from well-known examples such as the telephone, railroad, or highway systems. The latter are true *systems*, with one or a few basic functions, relatively stable properties, and readily identifiable boundaries. Inventors, "system-builders," and key organizations can be located and carefully studied. Comparative international studies of the same kinds of systems in different countries shed light on the social, political, and cultural shaping of those systems.[34]

In many cases, computer-based infrastructures originated as self-contained systems, such as the ERMA check-processing system, developed in the 1950s by the Bank of America, or the SABRE airline reservations system of the early 1960s, created by IBM and American Airlines. Competing systems then developed. This part of the story resembles the familiar history of, say, Bell Telephone and its early competitors in the 19th century. That competition was eventually resolved by the absorption of Bell's competitors and the conversion of the entire resulting system to Bell standards.[35]

In this initial phase, isolated information and control systems were computerized. By the middle 1960s, meta-level software allowed the exchange of data and programs between computer systems built by different manufacturers. However, the potential of this automatic interconversion was realized only later. Instead, "computer utilities" sprang up as the defining technology of their day. These were time-sharing mainframes, available for hourly rental. Users could share data with others on the same system or access a central software library, and they could take data or programs with them when traveling to other cities with branches of the same utility.[36] Although the utilities were known, in their time, as "networks," they were really no more than large systems in the Hughesian sense: they were relatively homogeneous, centrally controlled, and part of the unified vision of system builders at companies such as IBM, Tymshare, and University Computing Corporation.

In the 1970s, a fundamentally different form of integration began, well captured by the phrase "digital convergence." Translation software became available that could convert data from one format to another, while compilers allowed the same programs to run on different computers. Thus parties wishing to share data or programs no longer needed to choose between competing computer manufacturers or subscribe to computer utilities. Bell Laboratories' Unix – one of the first "portable," non-manufacturer-specific operating systems – became a particularly important vehicle for inter-system data exchange as it spread rapidly throughout the academic community.[37] The result was that many pre-existing systems survived to develop on their own, while *at the same time* being integrated through ever-larger networks.

These networks linked heterogeneous systems (including computers and software from multiple manufacturers), and network control was often distributed. Yet they did not fully escape the systems paradigm. One sign of this was their usually local nature. Local area networks, or LANs, within a single building were the first step. Large corporations extended these, over time, to cover regions or even larger areas, but connections among different corporate networks remained rare until the rise of the Internet in the latter half of the 1980s.

NETWORKS OF NETWORKS

Not itself a computer network, but a *network of networks*, the Internet connects a vast variety of local and wide-area computer networks. Internetworking exploited the potential of digital convergence through the simple, but elegant insight that the only requirement for connecting one network to another is a set of common protocols for data transmission. With internetworking, the number and heterogeneity of different systems that can be integrated becomes essentially unlimited, as does the possible range of system sizes.

From its earliest beginnings in the ARPANET of the late 1960s, the most fundamental principle of Internet design has been to assume heterogeneity in the networks being linked together. Internet protocols operate around and on top of existing networks, requiring from them very little, if any, internal change. If the Internet standard is not their native network data format, specialized computers or software can handle the conversion task. It is as if the railroads, rather than fixing a standard gauge (and thereby ending the useful life of undercarriages not built to that standard), had decided instead to build flexible track that could spread or shrink to accommodate every existing undercarriage – and every new one as well.

The basis for this massive interconnectivity is a set of protocols, or software and hardware standards, developed over three decades by an anarchical by surprisingly effective community of hackers and computer professionals. By analogy to Hughes' "system builders," we might call these people "protocol builders." Whereas the network builders who preceded them worked to ensure interconvertibility of data and programs within networks of heterogeneous computer systems, the protocol builders went one step further, creating techniques for exchanges among heterogeneous networks. Unlike the system builders, very few of the network builders nor the protocol builders have become well-known public figures, perhaps precisely because protocols are infrastructural, lying beneath the horizon of salient uses. Only recently has their hugely important role begun to be recognized.[38]

Although digitization removes limits on their extension, internetwork development patterns need not be computer-based. Consider, for example, transportation as a network of networks. Historically, moving goods from one place to another often required several different transport modes, such as rail, shipping, and trucking. Each of these modes could be considered a network of systems. Container shipping greatly improved efficiency, since it allowed goods to be moved from railroad cars to ships to trucks without unloading them – in effect, transforming three distinct networks into an internetwork.

Are internetworks simply systems of systems, in the Hughesian sense? I think not. Extending the transportation analogy may be useful in explaining this view. Many points in the total goods distribution network still require unit-by-unit unloading and reloading; only in the case of very large firms have these networks achieved the status of genuine, centrally controlled, well-bounded systems. Furthermore, the rail–ship–truck transportation network remains only imperfectly integrated with other globally important transportation modes, such as automobile, air, bicycle, foot, and animal-drawn systems. The physical differences between these modes make integration intrinsically difficult, if not impossible.

Consider the problems faced by metropolitan airport designers. Ideally, they must create smooth, efficient links among a wide variety of different transportation networks: automobiles, public buses, subways, occasionally even long-distance rail. Each of these carries its own set of constraints. All of these networks do the same thing, in a general sense, but at finer levels of detail the differences among them are enormous. (Readers who are regularly frustrated, as I am, by the waits, delays, and general chaos inherent in these intermodal connections will instantly appreciate the magnitude of this problem.) In addition, local geographies, economies, and many other factors require that every airport be specially designed. Thus airports are key nodes in a global transportation internetwork infrastructure – but they are not themselves "systems" in Hughes's sense of the term. Transportation is best characterized as an imperfectly integrated internetwork.

Table 1 summarizes some of the key distinctions I am trying to draw among systems, networks, and internetworks, as they apply to computer-based infrastructures.

These distinctions would need to be reinterpreted for the case of physical infrastructures. Nevertheless, they indicate that patterns

Table 1 Systems, Networks, and Internetworks

Systems	Networks	Internetworks
• one or several functions	• large number of functions	• near-infinite number of functions
• system builders (visionaries)	• network builders (interconversion techniques)	• protocol builders (standard interconversions)
• heterogeneous components, subsystems	• heterogeneous systems	• heterogeneous networks
• central control	• partially distributed control	• widely distributed control
• readily identifiable boundaries	• shifting boundaries	• continuous extension
• slow to moderate change	• potentially rapid change	• constant flux

discerned in the history of computer-based infrastructures might shed light on the history of technology in general.

WRITING THE HISTORY OF INFRASTRUCTURES

How can scholars make sense of the history of computer-based infrastructures? The vast heterogeneity of the systems in question makes it difficult even to get a grip on exactly what is important. Nevertheless, here I will attempt a sketch of how we might approach this daunting task.

On the technology side, a minimal list might include the study of patterns in the design and spread of the following:

- Timesharing systems. By the mid-1960s, these allowed multiple users to share a single mainframe and thus to communicate amongst themselves and share data on that machine.
- Compilers for high-level programming languages. These allowed different machines to run the same programs. An important effect was the creation of a market in software, which did not exist until the widespread adoption of FORTRAN and, especially, COBOL.
- Highly successful operating systems such as IBM's OS/360, Unix, Macintosh OS, and Windows.

- Networking hardware, such as Ethernet hubs. This became the basis for local computer networks fast enough to be truly useful, permitting the direct sharing of programs and data among many machines within a single building or a campus.
- Network and internetwork software protocols such as Ethernet and TCP/IP (Transmission Control Protocol/Internet Protocol, the fundamental Internet communication standard). Once established, these allowed different networks to be connected is a seamless web.

Still, these are merely some of the key building blocks of larger socio-technical systems and networks. Until we better understand the influences of social needs and political choices on the adoption and spread of these infrastructural technologies, we will not fully grasp their historical importance.

For example, the rapid adoption of personal computers in the corporate world, beginning around 1980, marked the demise of any possibility of system-building in the traditional sense. This phenomenon began in the "technological enthusiasm" of individual early adopters, most of them white-collar and middle-class. Initially, each bought his (or, sometimes, her) own computer, choosing whatever seemed most appealing from a wide array of *incompatible, stand-alone* machines and software. By 1985, many departments had introduced PCs on the same pattern (each unit choosing its own favorite, without regard for the rest of the company). Thus many large companies suddenly, within just two or three years, found themselves supporting dozens of different systems, mostly incapable of sharing data. As Gene Rochlin has remarked, attempts by managers to reassert centralized control over data and computing resources, after this chaotic start, were largely responsible for the rapid rise of networked systems and internets by the end of the 1980s.[39]

While the technical community that built the Internet is interesting and enormously important, socioeconomic factors like these – not technical capabilities alone – were the real force driving the global linkage of computing and communications. Castells's work represents one of the first attempts to connect the history of computer networks with the history of the development and spread of widely distributed, global corporations.[40] This is the sort of socio-technical analysis currently missing from most computer historiography.

Another way to approach the historiography of computer-based infrastructures would be functionalist in orientation. Rather than focus on machines, corporations, or software applications, one might imagine linked histories of the fundamental underlying capabilities of the modern computing paradigm. As noted above, these include the following:

- calculation
- communication
- control
- simulation
- information processing
- visualization

Of these, only calculation and information processing have yet received the exhaustive treatment they deserve.

This list might be expanded (albeit at risk of losing the macro-level view I am advocating) or revised. It could also be contracted. In my view, the three C's at the top of the list – *calculation, communication*, and *control* – are the most basic. A history that explained how these capabilities came to be so thoroughly intertwined and so widespread could be enormously enlightening.

For example, to my knowledge only a few scholars have systematically explored the development of computer control of other machines, and especially of other *systems*. David Noble's well-known account of the 1950s contest between record-playback and numerical (computerized) control of machine tools illustrated the complicated social ramifications of this fundamental technological change.[41] Programmable numerical-control tools made it possible for the designers of machined parts also to program their production – a process formerly under the control of skilled machinists. The choice for computerized control, promoted by the US Air Force and certain major industrial corporations, was part of a deliberate (and successful) effort to destroy the once-powerful machinists' unions in the context of the massive labor unrest that followed World War II. Shoshana Zuboff developed a similar line of reasoning in her longitudinal studies of computerization in a variety of work environments, from banks to paper mills.[42] She suggested that the computerization of control generated previously unavailable kinds and amounts of information about systems and their operators' performance. Zuboff referred to this quasi-natural outcome of computerization as the computer's power to "informate." For both Noble and Zuboff, the computerization of control systems changed the nature of work, reducing the importance of the tacit, bodily knowledge of manual operators while raising that of "intellective" (mental, symbol-handling) skill.

Yet these masterly examples merely scratch the surface of the history of computerized control. Since they treat particular sectors and cases, they never reach the infrastructure issues I am addressing.

Consider, for example, the problem of analog/digital conversion. In order for a computer to control another machine, the latter must be redesigned for digital input and output. The continuous (analog) signals generated by and used to control physical processes have to be changed into the discrete (digital) forms that computers can process, and vice versa. This has meant, in many cases, the adaptation of existing devices (e.g. sensors and actuators) or the construction of entirely new ones. Whole industries, such as digital compact disc recording and the numerical-control machine tools already mentioned, have been built around this change. From an infrastructure perspective, the fact that many such conversions were technically trivial does nothing to diminish their significance, since all of them added to the technological momentum required for digital convergence to become possible.

From the perspective of this essay, the crucial point here is that as machines were converted to digital control, ever larger *systems* of machines could be built, all controlled by a single computer. As Zuboff observed, this integration simultaneously made possible new understandings of overall system behavior, which in turn made possible increasing levels of integration. In the paper mills she described, operators of an entire plant spent most of their time in the "Star Trek Suite," monitoring machines and system flows from computer screens. Today, many integrated computer-controlled systems have grown far larger even than this. One, actually rather small example is the Upper Atmospheric Research Collaboratory. This Internet-based system gives scientists anywhere in the world real-time control of instruments located in the arctic in order to observe high-altitude phenomena, while sharing data and discussing results "live" on-line.[43] Another is the computerized telephone switching system that controls the large majority of telephone connections worldwide.

Similarly, most of the histories of which I am aware have very little to say about computers as a communication technology. This is not by any means identical with the history of computer networks, as I pointed out above. Only a historiography that places computers fully in the social, political, and cultural context will give this crucial development the attention it deserves.

One point of departure that reaches many infrastructure issues is the history of SAGE (Semi-automatic Ground Environment), the US continental air defense system built in the 1950s. SAGE was the first large-scale computerized control and communications system. From 23 "Direction Centers," the SAGE computers processed data from remote sensors (mostly radar), tracked incoming aircraft, and plotted interception trajectories for defending fighters and ground-to-air anti-aircraft missiles. In principle, the computers could even control the interceptors themselves, via their autopilots, as well as the release of weapons. The Direction Centers exchanged information automatically, over commercial telephone lines, using the first modems. The many military descendants of SAGE include the NORAD nuclear warning and control system, the World Wide Military Command and Control System, and the Vietnam-era Operation Igloo White. Even the Reagan-Bush "Star Wars" strategic Defense Initiative traced its lineage directly to concepts first developed for SAGE.[44]

These military systems are certainly the first, and best, examples of computer-based infrastructures integrating calculation, control, and communication functions.[45] The NASA telecommunications and weather satellite programs, which became important infrastructures in their own right, were probably the first major civilian analog of these, although computerized control of call switching in the Bell Telephone System could be another contender. Integrated infrastructures of this sort were slower to emerge in the commercial sector. There, private system-building was the focus for several decades.

Some sectors proceeded more rapidly than others in the development of computerized infrastructure. The financial industry, including banking and stock markets, may have been the first, with electronic funds transfer well-established by the end of the 1970s. In the Wall Street crash of 1987, the precipitous decline of stock prices caused the computerized trading systems at certain large financial institutions to initiate an automatic and massive sell-off, enormously aggravating the effects of the crisis. Gene Rochlin's recent book *Trapped in the Net* explores the unforeseen and the unforeseeable effects of automatic computer-based control and communications in emerging infrastructures. He, too, notes the dearth of studies of this phenomenon, one whose history and prospects we ignore at our peril.[46]

Recently, the President's Commission on Critical Infrastructure Protection (PCCIP) sounded a similar alarm, pointing to the rapid spread of "intermodal" infrastructures. For example, telephone systems, the Internet, and electric power grids are increasingly linked. With deregulation, electric power companies are turning to the Internet to keep track of supply and demand in a huge market with many suppliers. Many individuals rely on telephone lines for Internet access, and many Internet service providers are also, in fact, telephone companies such as MCI and AT&T. These intermodal links mean that a well-plotted terrorist attack on, say, a few major telephone switching centers and a few power transmission facilities might cripple all three systems at once. Similarly, a hacker could conceivably attack all three systems via the Internet, routing the attack through many countries around the world to prevent back-tracing of its source.[47]

Here at the turn of the millennium, the globalization of commerce and culture may be the single most important trend, full stop. It is one to which computers have contributed at least as much as anything else. To approach the history of computers as a history of infrastructure would give us a perspective commensurate with its scale and scope. Without it we are likely to remain, as one of Katie Hafner's interviewees aptly put it, "head down in the bits."[48]

Notes

1. M.S. Mahoney, "The History of Computing in the History of Technology," *Annals of the History of Computing* (1988) 10: 2, 113–125.
2. In computer jargon "K" is commonly used as an abbreviation for "Kilo" (i.e. the number 1,000, as in "kilometer").
3. I owe this insight to Michael Cohen of the University of Michigan School of Information. Cohen points out that the Y2K problem presents a unique opportunity to document and study the course of a technological crisis *before* it actually occurs. I have been unable to think of any other important technological breakdown with this unusual feature.
4. R. Chandrasekaran, "'Year 2000' Preview: Computer Flaw Already Bars Some Credit Card Processing," *Washington Post* (Date, 1998), A-01.
5. L. Wood, "VISA fines banks with Y2K problems," *RISKS-LIST: Risks-Forum Digest* (1997) 18: 74. Risks-Forum had been the world's major source for reports and discussion of important software problems for nearly twenty years. It exists only as an on-line discussion, moderated by Peter Neumann of SRI International. A searchable archive of all back issues is maintained at http://catless.ncl.ac.uk/ Risks.
6. J. Horner, "Re: Risks Associated with the Year 2000 Problem," *RISKS-LIST: Risks-Forum Digest* (1997) 18: 96.
7. C. Jones, *The Global Economic Impact of the Year 2000 Software Problem* (January 23, 1997) version 5.2, p. 27. Available for download at http://www.spr.com/html/ year_2000_problem.htm. Capers Jones is Chairman of Software Productivity Research, an international software consulting firm, and the author of numerous books and articles on software design and management.

8. E. Tenner, *Why Things Bite Back: Technology and the Revenge of Unintended Consequences* (New York: Knopf, 1996).
9. T.P. Hughes, *Networks of Power: Electrification in Western Society, 1880–1930* (Baltimore, MD: Johns Hopkins University Press, 1983); T.P. Hughes, "The Evolution of Large Technological Systems," in W. Bijker, T.P. Hughes, and T. Pinch, eds., *The Social Construction of Technological Systems* (Cambridge, MA: MIT Press, 1987), 51–82; T.P. Hughes and A.C. Hughes, eds., *Systems, Experts, and Computers* (Chicago: University of Chicago Press, forthcoming).
10. Official information on these initiatives may be found at http://www.iitf.nist.gov (the US Information Infrastructure Task Force) and www.gii.org (the Global Information Infrastructure Commission). Large archives of related materials, including excellent critical commentary, may be found at http://www.eff.org/pub/GII_NII (the Electronic Frontier Foundation) and http://nii.nist.gov (the US National Information Infrastructure Virtual Library).
11. P.N. Edwards, "Virtual Machines, Virtual Infrastructures: The New Historiography of Information Technology," *Isis* (1998) forthcoming.
12. H.D. Benington, "Production of Large Computer Programs," *Annals of the History of Computing* (1983) 5: 4, 350–361; T.J. Bergin and R.G. Gibson, eds., *History of Programming Languages II* (New York: ACM Press, 1996); J.E. Sammet, *Programming Languages: History and Fundamentals* (Englewood Cliffs, NJ: Prentice-Hall, Inc., 1969); S. Shapiro, "Splitting the Difference: The Historical Necessity of Synthesis in Software Engineering," *IEEE Annals of the History of Computing* (1997) 19: 1, 20–54; R.L. Wexelblat, ed., *History of Programming Languages* (New York: Academic Press, 1981).
13. J.J. Cortada, *The Computer in the United States: From Laboratory to Market, 1930 to 1960* (Armonk, NY: M.E. Sharpe, 1993); J.W. Cortada, *Information Technology as Business History* (Westport, CT: Greenwood Press, 1996).
14. Rob Kling's Social Informatics website (http://www-slis.lib.indiana.edu/SI/index.html) is a good source of information on this perspective, as is the journal *The Information Society* (http://www-slis.lib.indiana.edu/TIS/index.html), published since 1981.
15. Good examples of this approach are J. Weizenbaum, *Computer Power and Human Reason: From Judgment to Calculation* (San Francisco: W.H. Freeman, 1976); T. Roszak, *The Cult of Information* (New York: Pantheon, 1986).
16. R. Kling and S. Iacono, "Computing as an Occasion for Social Control," *Journal of Social Issues* (1984) 40: 3, 77–96; R. Kling and W. Scacchi, "The Web of Computing: Computing Technology as Social Organization," *Advances in Computers* (1982) 21: 3–85.
17. D. Hakken, *Computing Myths, Class Realities: An Ethnography of Technology and Working People in Sheffield, England* (Boulder, CO: Westview Press, 1993); R. Kling, "Computerization and Social Transformations," *Science, Technology, & Human Values* (1991) 16: 3, 342–367; T.K. Landauer, *The Trouble with Computers: Usefulness, Usability, and Productivity* (Cambridge, MA: MIT Press, 1995); D. MacKenzie, "The Influence of the Los Alamos and Livermore National Laboratories on the Development of Supercomputing," *Annals of the History of Computing* (1991) 13: 2, 179–201; L. Suchman, *Plans and Situated Actions: The Problems of Human-Machine Communication* (New York: Cambridge University Press, 1987); S. Turkle and S. Papert, "Epistemological Pluralism: Styles and Voices within the Computer Culture," *Signs* (1990) 16: 1, 128–157.
18. P.N. Edwards, *The Closed World: Computers and the Politics of Discourse in Cold War America* (Cambridge, MA: MIT Press, 1996); K. Flamm, *Creating the Computer: Government, Industry, and High Technology* (Washington, D.C.: Brookings Institution, 1988); H.D. Hellige, "From SAGE via ARPANET to Ethernet: Stages in Computer Communications Concepts between 1950 and 1980," *History and*

Technology (1994) 11: 49–75; K.C. Redmond and T.M. Smith, *Project Whirlwind: The History of a Pioneer Computer* (Boston: Digital Press, 1980).

19. D.S. Bushnell and V.B. Elder, "Computers in the Public Interest: The Promise and Reality of ARPANET," in J.P. Jacky and D. Schuler, eds., *Directions & Implications of Advanced Computing* Vol. I (Norwood, NJ: Ablex Publishing Corp., 1989), 33–50; K. Hafner, *Where Wizards Stay Up Late: the Origins of the Internet* (New York: Simon and Schuster, 1996); H.D. Hellige, "From SAGE via ARPANET to Ethernet: Stages in Computer Communications Concepts between 1950 and 1980," *History and Technology* (1994) 11: 49–75; A.L. Norberg and J.E. O'Neill, *Transforming Computer Technology: Information Processing for the Pentagon, 1962–1986* (Baltimore: Johns Hopkins University Press, 1996).

    Another interesting source on ARPANET/Internet history is R. Hauben, *The Netizens' Netbook* (1996), http://www.columbia.edu/~hauben/netbook/. This is a Web-based history of the Internet and World Wide Web. Like other Web-based materials, it undergoes continual revision.
20. G.D. Crowe and S.E. Goodman, "S.A. Lebedev and the Birth of Soviet Computing," *IEEE Annals of the History of Computing* (1994) 16: 1, 4–24; N.C. Davis and S.E. Goodman, "The Soviet Bloc's Unified System of Computers," *ACM Computing Surveys* (1978) 10: 2, 93–122; G.L. Geipel, A.T. Jarmoszko, and S.E. Goodman, "The Information Technologies and East European Societies," *East European Politics and Societies* (1991) 5: 3, 394–438; S.E. Goodman, "Socialist Technological Integration: The Case of the East European Computer Industries," *The Information Society* (1984) 3: 1, 39–90; W.K. McHenry and S.E. Goodman, "MIS in Soviet Industrial Enterprises: The Limits of Reform from Above," *Communications of the ACM* (1986) 29: 11, 1034–1043.
21. E.D. Langer, "Generations of Scientists and Engineers: Origins of the Computer Industry in Brazil," *Latin American Research Review* (1989) 24: 2, 97–111; E.M. Roche, "Brazilian Informatics Policy: The High Costs of Building a National Industry," *The Information Society* (1990) 71–32.
22. A.D. Chandler, *The Visible Hand: The Managerial Revolution in American Business* (Cambridge, MA: Belknap Press, 1977).
23. For an insightful analysis of macro-level approaches, see T.J. Misa, "Retrieving Sociotechnical Change from Technological Determinism," in M.R. Smith and L. Marx, eds., *Does Technology Drive History?* (Cambridge, MA: MIT Press, 1994), 115–141.
24. J.R. Beniger, *The Control Revolution: Technological and Economic Origins of the Information Society* (Cambridge, MA: Harvard University Press, 1986).
25. M. Castells, *The Rise of the Network Society*, Vol. 1 of *The Information Age: Economy, Society and Culture* (Cambridge, MA: Blackwell Publishers, 1996); M. Castells, *The Power of Identity*, Vol. 2 of *The Information Age: Economy, Society and Culture* (Cambridge, MA: Blackwell Publishers, 1997).
26. M. Castells, *The Informational City: Information Technology, Economic Restructuring, and the Urban Regional Process* (Cambridge, MA: Blackwell, 1989), pp. 10–12.
27. Castells, *Network Society*, esp. pp. 32–65.
28. Indeed, Hughes' latest book, co-edited with Agatha Hughes, is titled *Systems, Experts, and Computers* (University of Chicago, in press). It includes chapters on a wide variety of arenas for which systems thinking, information theory and metaphors, and computers became central in the decades after World War II.
29. W. Bijker, T.P. Hughes, and T. Pinch, eds., *The Social Construction of Technological Systems* (Cambridge, MA: MIT Press, 1987); W.E. Bijker, "Sociohistorical Technology Studies," in S. Jasanoff, T. Pinch, G. Markle, and J. Petersen, eds., *Handbook of Science and Technology Studies* (Beverly Hills, CA: Sage Publications, 1994), 229–256; R. Mayntz and T.P. Hughes, eds., *The Development of Large*

Technical Systems (Boulder, CO: Westview Press, 1988); G. Hecht, *The Radiance of France: Nuclear Power and National Identity after World War II* (Cambridge, MA: MIT Press, forthcoming, 1998).

30. One example of this rapprochement is the Sloan Foundation-funded project for "a new American history textbook that seeks to incorporate science and technology into the narrative of the nation's history," spearheaded by historians of technology Merritt Roe Smith and Daniel Kevles. Sloan Foundation Public Understanding of Science and Technology Program, http://www.sloan.org/education/Understanding_Sci&Tech.html.
31. The social side of digital convergence is reflected in, for example, the phenomenon of people replacing their entire collections of analog music recordings (on magnetic tape and vinyl disc) with digital CDs. For most people's purposes, there was nothing particularly wrong with the older analog format, and in fact many audiophiles still insist that analog recordings are purer and more accurate than digitized ones. Yet the prestige of the new digital format acquired such momentum that an entire generation of audio equipment became obsolete within a decade. Similar things could be said about other art forms, such as photography.
32. The current fascination with computerization has led designers to introduce digital formats in many applications where they are completely inappropriate. Good examples are automobile speedometers and stereo systems, where digital readouts and digitzed controls are far more cumbersome than their traditional analog counterparts.
33. Misa, "Retrieving Sociotechnical Change."
34. Mayntz and Hughes, *Development of Large Technical Systems*: Bijker, "Sociohistorical Technology Studies."
35. L. Galambos, "Looking for the Boundaries of Technological Determinism: A Brief History of the U.S. Telephone System," in Mayntz and Hughes, *Development of Large Technical Systems*, 135–153.
36. Campbell-Kelly and Aspray, *Computer*, pp. 215–219.
37. See Chapter 9 of Hauben, *Netizens' Netbook*, 6/15/96 version, entitled "On the Early History and Impact of UNIX: Tools to Build the Tools for a New Millennium."
38. Hafner and Lyon, *Wizards*; Norberg and O'Neill, *Transforming Computer Technology*.
39. G.I. Rochlin, *Trapped in the Net: The Unanticipated Consequences of Computerization* (Princeton, NJ: Princeton University Press, 1997).
40. Castells, *Network Society*.
41. D. Noble, *Forces of Production* (New York: Oxford University Press, 1984).
42. S. Zuboff, *In the Age of the Smart Machine: The Future of Work and Power* (New York: Basic Books, 1988).
43. http://www.crew.umich.edu/UARC/
44. Edwards, *Closed World*.
45. Hellige, *From SAGE via ARPANET*.
46. Rochlin, *Trapped in the Net*.
47. All documents of the PCCIP, including the final report, may be found at http://www.pccip.gov/index.html.
48. Hafner and Lyon, *Wizards*.