"The Hard Work of Software History"

Henry Lowood

Curator for History of Science & Technology Collections, Stanford University Libraries

RBMS Pre-Conference, "The Twentieth Century"

San Francisco, June 13, 2001

A few years ago, the literary and media historian Friedrich Kittler opened an essay called "There Is No Software" with a "rather sad statement." In his view, "the bulk of written texts – including this text – do not exist anymore in perceivable time and space but in a computer memory's transistor cells." Coming from a scholar who had until then situated the cultural meaning of literary texts in discourse networks dependent on technologies of inscription (writing, gramophone, typewriter, computer) and the materiality of communication, this remark captures the essence not just of a technological change but of a significant cultural shift. At the end of the 20th century, according to Kittler, texts – and even software itself – have vanished. Our text-producing gestures merely correspond to codes built on silicon and electrical impulses; the texts themselves no longer exist materially, and indeed we have ceased to write them: "All code operations … come down to absolutely local string manipulations and that is, I am afraid, to signifiers of voltage differences."[1] Following Kittler's train of thought, we should wonder how libraries and archives will locate electronic or virtual replacements for the acts and artifacts of writing that occupied Goethe at the turn of the 18th century or Einstein at the close of the 19th.

The impact of software extends beyond the replacement of paper-based media, of course. Software has become a condition of our lives so many ways that it has become part of the environment, making it increasingly difficult to recognize what is significant to preserve. About ten years ago, the computer scientist Mark Weiser described the then future omnipresence of software in an article titled "The Computer for the 21st Century" published in *Scientific American*. This essay introduced Weiser's research program, which he dubbed "ubiquitous computing," to this magazine's technologically literate readership, eager to read about plausible visions of the future. Much of Weiser's argument hinged on a straightforward observation, but one that nonetheless turned his views in an unexpected direction: "the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."[2] In the first historical phase of computing as he saw it, many people shared one large computer, such as a time-shared IBM mainframe, the Big Iron of the Information Age. Then with the advent of the microprocessor, computers became personal: one person, one machine. In his work at Xerox Palo Alto Research Center until his premature death two years ago, Weiser – whose papers are now at Stanford – created small, portable, and networked devices for times in which computers would far outnumber people, the third age of ubiquitous computing. He believed it significant not that computers would outnumber people, but that they would have to become "invisible" in order to become useful. As he phrased it a few years after publishing the *Scientific American* article, the "highest ideal is to make a computer so imbedded, so fitting, so natural, that we use it without even thinking about it."[3] Indeed, Weiser often referred to the Third Age of Computing as "the age of calm technology," meaning that ubiquitous computers would become unremarkable elements of our surroundings, neither threatening nor interfering with our daily activities.

The insights of Kittler and Weiser offer different but reinforcing versions of the disappearance of software in a world of computers. Both recognize that profound cultural changes of the last two or three decades can be credited to the impact of computing, so much so that software has become ubiquitous, even "invisible." The notion that computers have taken over our lives had already become fairly commonplace in the 1990s, but at a deeper level, Kittler and Weiser identified transformations that had become sources of *malaise* to some and exhilaration to others; future historians may regard both as fundamental aspects of our civilization and see both as centered on the notion that the media of our culture, whether text or technology, are no longer found in its material traces but rather in the imperceptible, the virtual, and the invisible—in short, in software.

My topic today is the challenge these historians will face in documenting cultural and technological changes that by their very nature have transformed the substance of historical documentation and radically altered the conditions of its preservation. For the most part, I will concentrate on the cultural medium of this transformation – software – and its history. By software, let it be understood that I am using the term loosely to include not just code and executable programs, but also digital media dependent on software, and, at times, computing generally. I will also comment here and there on the changes that efforts to preserve the history of software may impose on institutions such as libraries, archives, and museums.

My Stanford colleague Tim Lenoir has written that he is "intrigued by the notion that we are on the verge of a new renaissance, that, like the Renaissance of the fourteenth and fifteenth centuries, is deeply connected with a revolution in information technology." He describes the transformation of our times as "heralding a posthuman era in which the human being becomes seamlessly articulated with the intelligent machine." [4] Some of you in this audience might be more comfortable with the "printing revolution in early modern Europe," the title of Elizabeth Eisenstein's now famous book, than you are with the notion that texts, technology, and even humanity have become dependent upon, even integrated with, computer-based information technologies. On the other hand, who can better appreciate the intellectual issues raised by profound transformations of media than you, the historians and conservators of print and manuscript culture?

In my brief tour through some of the pitfalls and possibilities in building software history collections, I will begin with a short introduction to the history of software as a medium and describe a few characteristics of software that are likely to be important in historical perspective. From this wobbly ledge, I will dive into the turbulent sea of problems that obscures the potential treasure of historical software collections. After drying off with a few examples of what has nonetheless been accomplished thus far, I will finish this talk with a cold shower by considering how providing access to these collections will raise organizational issues for archives, libraries, and museums.

The History and Historiography of Software

In light of the dependence of software on hardware, we should not be surprised that most histories of the software industry begin with their separation.[5] The short version of this story takes off from the announcement by IBM in June 1969 that it would un-bundle the provision of software from the sale or lease of its computer systems. In other words, until 1969 most software came bundled with computer hardware systems, the very industry dominated by IBM. Not that independently developed and marketed software was completely unknown, but it was largely limited to special-purpose applications or academic projects.

During the 1970s, the business, culture, and technology of software production changed dramatically. Of course, the industry grew rapidly after 1969. According to Martin Campbell-Kelly, sales of software in 1970 represented less than 4 percent of the entire computer industry. The volume of sales increased from this base more than twenty-fold by 1982, fifty-fold by 1985.[6] At about the same time, the term software engineering took hold to describe systems of software production based on theories and methods of computer science, stimulated by the first NATO Conference on Software Engineering in 1968. The proponents of software engineering applauded the establishment of computer science as a legitimate scientific field. The first academic departments in this new discipline were founded at institutions such as Purdue and Stanford in the early to mid-1960s; these new departments shifted the weight of attention to the study of software techniques, as opposed to the hardware engineering already sufficiently represented in electrical engineering and applied physics.

By the end of the 1960s, the very meaning of software was also evolving. Fundamental innovations in interface design and electrical engineering provided new platforms for changing the nature of computing and redefining software. Douglas Engelbart's work at the Stanford Research Institute, for example, liberated the computer from its primary role as a calculating engine and headed it toward a future centered on information management and networked communications. The system designed by Engelbart and his team of programmers at SRI's Augmentation Research Center debuted spectacularly at the 1968 Fall Joint Computer Conference held at the San Francisco Convention Center, just a few blocks from here. The legendary demonstration inspired a generation of computer scientists to dream of new systems replete with mice, windows, icons, and desktops. Only a few months later, Stanley Mazor, Ted Hoff, and Federico

Faggin designed the first single-chip Central Processing Unit—a computer for all intents and purposes—which Intel introduced to the world as the 4004 microprocessor in 1971. Within a few years, microprocessors made microcomputers possible, setting the stage for the rapid development of the personal computer during the 1980s and new generations of software and computer interfaces.

Until these developments, the creation of software was inextricably tied to the relatively closed world of computer engineering. In time, the corporate world of Big Blue (IBM) gave way to the computers "for the rest of us," a change immortalized in the famous Macintosh Super Bowl advertisement of 1984. Writing on the history of software production lagged behind these changes and focused until recently on the period from roughly 1945 to 1970, and thus for the most part followed early hardware development. Paul Edwards, one of a new generation of historians of computing, situates the older historiography in what he calls the tradition of "machine calculation," while also locating a distinct set of writings and historical actors in a separate tradition of "machine logic" (software). The ancestry of this latter tradition, he argues, "lies in mathematics and formal logic."[7] In his book on the "Closed World" of Cold War computing, Edwards observed that historical accounts within these internalist traditions rarely have ventured beyond the perspectives of those scientists and engineers whose technical achievements defined them. Further, again according to Edwards, "There is little place in such accounts for the influence of ideologies, intersections with popular culture, or political power."

Changes in the Historiography of Software

Like software, the historiography of software has redefined itself. As noted already, changes in the industry, technology, and culture of computing from the 1970s to the present shifted the aspirations of software designers and programmers. As historians have begun to come to grips with these changes, limitations in the historiography of "machine calculation" and "machine logic" have become more apparent. Wider, then widespread access to computer technology has intensified interest in the social, cultural, and business history of computing, topics of no little importance for a new social construction of software. The "PC Revolution" of the late 1970s and early 1980s revealed intersections among the contributions of computer scientists, software engineers, hobbyists, and entrepreneurs. They appeared in the founding of organizations such as the Homebrew Computer Club and the People's Computer Company, the titles of books such as Ted Nelson's *Computer Lib/Dream Machines,* first published in 1974, and the rise of companies like Atari and Apple Computer. Douglas Engelbart, Ted Nelson, Alan Kay and others active in the 1960s and 1970s set the stage for the rapid development of software technology in the 1980s and beyond. A few examples are the role played by Nelson's hypertext in the creation of the World Wide Web, the influence of Engelbart's SRI lab and Kay's work at Xerox PARC on the development of graphical user interfaces such as those embedded in the Macintosh and Windows, and the many spinoffs of Cold War research in artificial intelligence and other areas of computer science for information technologies such as library catalogs. An authoritative history of software since the late 1960s has not yet been written, but when it is, its author will face the task of synthesizing a rich and variegated history extending beyond the internal development of code, languages, and protocols.

The Difficulties of Collecting Software

These brief remarks on the history of software merely set the stage for considering the difficulties – some would say the impossibility – of collecting software. Archivists, librarians, curators, and historians today face the daunting task of documenting strands of software history such as those briefly noted above, including their cultural impact, and providing source materials for studying them. In the first instance, this means sifting through the virtual mountain of electronic media and information that has grown around us in the last three decades. Of course, there is more to it than that. The expenditure of resources to create archives of software is difficult to justify without first considering how historians ten or a hundred years from now might reflect upon this ubiquitous but invisible technology, as well as speculating how they might have access to it. The emancipation of software production from the closed and bundled world of computer engineering since the 1970s has rapidly accelerated our dependence on software, increasing not only our interest in its historical development but also our awareness of the evolving nature of information resources and storage. The new developments of the last quarter-century, including personal computer technology, graphical interfaces, networking, productivity software, electronic entertainment, the Internet, and the World Wide Web, have expanded the use of software and profoundly altered the discourse of software history, while at the same time delivering an astonishing potential wealth of electronic data for historical analysis.

Future historians of the last three decades are sure to study the dizzying rate of change in the uses of software for supporting new media of communication, entertainment, and information management. Recall that the research of Engelbart, Nelson, and others established the computer as a communication machine and reversed its prior meaning as primarily a calculation engine. This monumental expansion of the nature of software has been followed by convergences of media and software technology that will push software historians into nearly every medium of entertainment, art, story-telling, and information management. Software has become many things to many people, occupying the work, leisure, and creative time of millions of non-programmers as well as software designers.

The broader social and cultural impact of computing will – if it has not already – revolutionize all cultural and scholarly production. It follows that historians (not just of software and computing) will need to consider the implications of this change, and they will not be able to do it without access to our software technology and what we did with it. Software and digital information have begun to rival printed materials, visual media, and manuscripts as primary sources in many fields of inquiry, while writers, artists, musicians, game designers, and even historians work productively in the media of computing.

Often, every trace of these activities, save our memory of them, has been born digital. Consider one example from my personal experience: I recently attended the conference of the Electronic Entertainment Exposition in Los Angeles as a session panelist. E3, as it is known, is the Mecca of computer-based entertainment, and it draws the digital generation like journalists to a free lunch (which, in fact, is one of the attractions). The organizer of my panel on "Computer and Console Games – A Cultural Legacy?," Justin Hall, explained to me over lunch on the second day how he had already sold to the highest bidder an article written the night before our panel. Between bites of sandwich he tapped away at his laptop keyboard and downloaded images from his digital camera, but only after returning home from the conference did I learn that everything about this transaction, as well as his text, was already embedded in silicon, as Kittler might say, and available for me to view. Hall has been keeping an online, web-based diary continuously since 1994, "Justin's Links from the Underground."[8] I checked this source and found an entry matching the day we had lunch: "Bid on my Article: E 3 Way to claim it for your web site. Highest bidder at midnight, Pacific time, gets the article and an exclusive photo for their site or magazine!" This web link in this sentence led to a completed eBay auction, from which I learned the price of the article ($14.50) and that the winning bidder was aleonard, presumably Andrew

Leonard of salon.com. After a moment's further web browsing, sure enough, the article could be found on the salon.com website.[9] A moment's of further browsing, and I found the article at salon.com. In other words, this article was written, photographed, sold, and published without a single "written," or paper, trace.[10] And yet, the article and interesting details about the transaction were readily available. Of course, there are millions of similar examples of commerce, entertainment, authorship, artistic creation, journalism, science, and even software engineering carried out without paper. Each one adds to the urgency of software preservation, digital archiving, and accessible electronic libraries on a front far broader than the history of computing.

The relentless advance of computer technology on an ever-expanding set of fronts is redefining the nature and scope of computing itself. It could be argued, at least from the vantage-point of the present, that human beings interact directly with computers more than with any other technology. In many contemporary families, computers have partly replaced television sets, radios, and telephones. In *The Road Ahead,* published in 1995, Bill Gates provided a vision of the near future of computing that explicitly includes *all* "mediated experiences," whether of commerce or culture. Historians of software, clearly, will have to venture into every niche, nook and cranny of society in ways that will separate their work from other historians of science and technology. It has become far more difficult to locate the edges of computing as a discipline and to map the boundaries of its impact on society than for most other technical and scientific fields. The open-ended nature of computing challenges archivists, librarians, and curators, and it complicates matters for researchers looking for disparate materials in a variety of media and repositories.

So what do we do in the face of the growing volume, diversity, and importance of software? Part of the difficulty in defining next steps is that the very cat we are trying to put in the bag is ripping all of our heirloom luggage to shreds. This is perhaps where the history of software least resembles the history of print culture. This is not so much in the impermanence of its media – an issue upon which the dust has not yet settled – but in the flexibility of its use, with the capacity for converging previously separable realms concerned with what we now call "content": texts, stories, audio-visual experiences, interactive simulations, data processing, records management, and metadata applications such as indexing, among them. Traditional institutions and professional identities provide uncertain guidance in deciding who is responsible for the custodial care of software, given this diverse range of applications and associated knowledge. As Doron Swade points out from the perspective of a museum curator:

"Some software is already bespoke: archivists and librarians have 'owned' certain categories of electronic 'document': Digitised source material, catalogues, indexes, and dictionaries, for example. But what are the responsibilities of a museum curator? Unless existing custodial protection can be extended to include software, the first step towards systematic acquisition will have faltered, and a justification for special provision will need to be articulated ab initio in much the same way as film and sound archives emerged as distinct organisational entities outside the object-centred museum."[11]

Swade considers the problem as one of "preserving information in an object-centred culture," the title of his essay; that is, he ponders the relevance of artifact collections of software and the various methods of "bit-perfect" replication of their content. Libraries, and within libraries rare books and manuscript librarians, are coming to grips with related issues that might be described as "preserving information in a text-centred culture." In saying this, I realize that exactly these librarians are often the chief protectors of artifact-centered culture in American libraries. Nonetheless, their *raison-d'être* is the preservation of special categories of original source materials – primarily texts -- for programs of academic research and teaching. This is one of the rubs in formulating institutional approaches to the preservation of software and related digital media, for software defines a new relationship between media objects and their content, one that calls into question notions of content preservation that privilege the original object. Current debates about the best methods for preserving software, which I have no intention of rehearsing here, are partly stuck on different institutional and professional allegiances to the preservation of objects, data migration, archival functions, evidentiary value, and information content. I fear that these issues are not likely to be sorted out before it is necessary to make serious commitments at least to the stabilization, if not the long-term preservation, of digital content and software. Projects like Brewster Kahle's Internet Archive have demonstrated what it is already possible to accomplish.[12]

What Can Be Done? Some Projects and Programs

Preservation of the records of software history has benefited from archival and historical work in other areas. By the late 1970s, archival organizations, historical repositories, and professional societies had begun to pay systematic attention to the history of recent science and technology. Disciplinary history centers such as the American Institute of Physics (AIP) History Center, the IEEE History Center, and the Charles Babbage Institute were established in part to coordinate and support the preservation of historical documentation and to work with existing repositories to address issues of archival appraisal, preservation, and access. In the early 1980s, the Society of American Archivists, History of

Science Society, Society for the History of Technology, and the Association of Records Managers and Administrators co-sponsored a Joint Committee on Archives of Science and Technology, known as JCAST. Its report, *Understanding Progress as Process: Documentation of the History of Post-War Science and Technology in the United States*, represented an important milestone when published in 1983, especially by raising awareness among American archivists of their need to understand better the records of post-war science and technology.

A loosely knit group of archival repositories and, just as important, an evolving set of principles and practices emerged out of archival research and projects like the JCAST report. Guidelines for appraisal of records and documentation strategies set the stage for projects. By the late 1980s, the first published guides to collections in the history of computing appeared in print: *Resources for the History of Computing,* edited by Bruce Bruemmer, *The High-Technology Company: A Historical Research and Appraisal Guide* by Bruce Bruemer and Sheldon Hochheiser, both published by the Babbage Institute, and *Archives of Data-Processing History: A Guide to Major U.S. Collections,* edited by James Cortada and published by Greenwood Press. Together, they effectively document the strategies and programs that guided the growth of archival resources in the history of computing up to about 1990. Yet, it was clear that the work had only begun. Cortada noted that:

"The first group of individuals to recognize a new subject area consists usually of participants followed closely after by students of the field and finally, if belatedly, by librarians and archivists. It is very frustrating to historians of a new subject, because it takes time for libraries to build collections or to amass documentary evidence to support significant historical research. This situation is clearly the case with the history of information processing."[13]

During these initial stages, the archival records and documentation available on the history of computing was largely paper-based. The establishment of the archives of the Charles Babbage Institute at the University of Minnesota in 1979 – the CBI had been founded at Stanford a few years earlier – was a signal event in this phase. Symbolically, so was the publication of a brochure on behalf of the History of Computing Committee of the American Federation of Information Processing Societies (AFIPS), called "Preserving Computer-Related Source Materials" and distributed at the National Computer Conference that year. The information in this brochure was inspired by the accomplishments of the Center for the History of Physics of AIP, and it recommended that:

"If we are to fully understand the *process* of computer and computing development as well as the end results, it is imperative that the following material be preserved: correspondence; working papers; unpublished reports; obsolete manuals; key program listings used to debug and improve important software; hardware and componentry engineering drawings; financial records; and associated documents and artifacts."[14]

The text focused almost entirely on the preservation of paper records as such, even printouts, manuals, and text listings of programs, but nowhere mentioned the preservation of data files, merely noting with a nod to the museum value of hardware artifacts that "Actual computer componentry is also of great interest. The esthetic and sentimental value of such apparatus is great, but aside from this, the apparatus provides a true picture of the mind of the past, in the same way as the furnishings of a preserved or restored house provide a picture of past society."[15]

Even in the absence of a mandate to save software, libraries, archival repositories, and museums have mobilized resources to document the history of computing. Historians of software will draw on a variety of historical documentation that includes many formats, both digital and paper-based. Due to the widening realm of software applications, hundreds, if not thousands, of repositories have saved – sometimes inadvertently – software itself or materials that inform us about contexts of its creation and use. Consider topics such as the history of hospital information management, library database technology, scientific computation, digital typography, or computer graphics in the film industry, topics for which documentation may be found in repositories ranging from government record centers and university archives to closed private collections and corporate records centers. The spectrum of institutions holding materials of software history is virtually without limit, especially with the inclusion of truly "virtual" collections such as Brewster Kahle's Internet Archive.

Following Weiser, perhaps the omnipresence of software has led us to become overly calm about its preservation, since few institutions have explicitly taken up the challenge. *Archives of Data-Processing History* provided a good overview of the major repositories in the field circa 1990, and this circle has not widened considerably since that time, even though many collections have been added since then. The core group of bricks-and-mortar collections consists of the Charles Babbage Institute, the Computer Museum (now the Computer History Center), the Hagley Museum and Library, the Library of Congress, the National Archives and Records Administration, the Smithsonian Institution, and the Stanford University Libraries, plus several corporate archives (IBM, AT&T, Texas Instruments, etc.). Smaller, but nonetheless significant collections can be found in university libraries and archives at Dartmouth, Harvard, MIT, Carnegie-Mellon, Illinois, and Pennsylvania as a consequence of the historical role of these institutions, rather than active collecting programs. In short, there are certainly fewer than ten institutions in the United States that actively collect research materials in traditional formats for the history of computing. Growth since the early 1990s in available documentation has occurred largely as a result of independent, largely web-based initiatives, such as the RFC (Request for Comment) Index of key documents on the development of the Internet, private initiatives such as the Internet Archive, and many other collections of digitized and born-digital materials assembled and accessible via online archives, home pages, and corporate websites. In a sense, a second generation of software archives has emerged in its own medium, creating a recursive problem concerned with the long-term preservation of these digital archives.

The Stanford University Libraries, where I have been curator of the history of science and technology collections since 1983, maintains an active archival program in the history of computing. Let me take a few minutes now to use our program as an example for how institutions go about acquiring collections of historical records relating to software. The Stanford Libraries' program in the history of computing grew on two legs: first, an archival orientation in the narrow sense, focused on records of activities that took place at Stanford, and, second, a collecting program founded in 1984 and called the Stanford and the Silicon Valley Project, today known as the Silicon Valley Archives. The idea behind the Silicon Valley Project was straightforward: Compile documentation tracing relationships connecting Stanford faculty and graduates to emerging high-technology industries in the surrounding region since the 1930s. It extended a flourishing program in the University Archives that, by the mid-1980s, had assembled collections of faculty papers and university records in the sciences and engineering. For software history, relevant collections in the University Archives include the papers of Ed Feigenbaum, John McCarthy, George and Alexandra

Forsythe, Donald Knuth, and many others, as well as records of the Center for Information Technology (Stanford's computation center), the BALLOTS project papers (an early project in the area of library automation and database technology), the ACME Project collection (a collaboration of Edward Feigenbaum and Joshua Lederberg that led to path-breaking software in the field of expert systems such as MYCIN and DENDRAL), and the Heuristic Programming Project. As the Department of Computer Science, founded in 1965, has become perhaps the leading university program in its field, the University Archives has, by preserving records of its programs and faculty papers, grown in importance for the history of computing.

By 1984, it had become clear not only that the explosive growth of Silicon Valley dominated regional development, but that it was also a forerunner of other highly concentrated techno-scientific regions. Due to the close connections between Stanford and specific business ventures located in Silicon Valley, the University Archives already owned significant collections relevant to its historical development. It was a logical step for the Department of Special Collections and University Archives to move forward and actively collect records of Silicon Valley enterprises and individuals not directly tied to Stanford. It appeared that no other institution would invest resources to locate and preserve archival materials documenting research and business growth characteristic of Silicon Valley industries. As a result of the institutional decision to move forward, Stanford has acquired substantial company and laboratory records, such as those of Fairchild Semiconductor Corporation, the American Association for Artificial Intelligence, the System Development Foundation, SRI laboratories under the direction of Douglas Engelbart and Charles Rosen, Mark Weiser's work at Xerox PARC, Interval Research Corporation and Apple Computer. Once the parameters of our project had been established, we proceeded to work with faculty who were known to have contacts in Silicon Valley industry, such as Edward Feigenbaum and, more recently, Doug Brutlag. Another vector from Stanford out to Silicon Valley along the path of software, one of particular interest to this meeting, has been followed in digital typography, with the acquisition of the Euler Project papers of Hermann Zapf and the voluminous papers of Donald Knuth.

A new twist in the story of the Silicon Valley Project has been the collecting of computer software and, to a lesser extent, hardware. The urgency of these efforts has been intensified by research projects that seek to tell the story of the Silicon Valley in its own medium. In the first instance, Stanford has acquired materials such as data-tapes from the Augmentation Research Center at SRI and Engelbart's projects there, hard disk images accompanying collections of personal papers such as those of Jef Raskin and Mark Weiser (not to mention those now frequently acquired with literary papers), e-mail archives, streamed media and digitally taped audio- and video-interviews, electronic versions of student papers, and packaged commercial software, including thousands of titles in the Stephen F. Cabrinety Collection in the History of Microcomputing, which includes one of the world's largest collections of early computer and video games. Each of these formats requires special strategies for evaluating, recovering, stabilizing, possibly reformatting, and indexing content. For the most part, tested strategies do not yet exist; in a few cases, we have embarked on our own special projects to test techniques for insuring that future historians will have access to the contents of software collections. For example, in the case of computer game software, Tim Lenoir and I are heading a project called "How They Got Game: The History and Culture of Computer Games and Simulations," funded by the Stanford Humanities Laboratories. As part of this project, the results of which will appear entirely in electronic form, we are evaluating a three-pronged approach to the documentation of game software: streamed video of gameplay, location and preservation of source code, and scanned images of related packaging, marketing materials, and documentation. Note that our efforts thus far have steered relatively clear of emulators, meta-data packaging, and the preservation of hardware, techniques currently at the center of contention among museum curators, archivists, and librarians about best practices for long-term preservation of digital documents. While there is great potential for useful work, say, in emulator development, our point is that even in the absence of a final verdict on these strategies, it is still possible to create useful software history resources that can be preserved.

Institutional Issues

Although I have certainly left out more topics than I have covered, I would like to conclude now with a few remarks about the role of Special Collections in the preservation of software history. As we have seen, both at Stanford and at other institutions, the archival impulse in the history of computing began with paper-based records and documentation. The printed guides cited earlier list the personal papers of computer scientists held in manuscript collections and archives, oral histories, and corporate records. Early computers have been saved by museums such as the Computer History Center and the Science Museum in London, and libraries have saved collections of documentation, technical reports, and the early computing literature. At Stanford, as elsewhere, manuscript, ephemera, and, to a lesser extent, book collections in the history of computing have landed in Special Collections and University Archives as an extension of earlier patterns of collecting practice. As the nature of this documentation shifts from paper to electromagnetic storage media, issues of access and technological complexity are calling this comfortable habit into question.

Access to software collections is the first problem. The mission of departments of special collections, especially in university libraries, includes not just preservation but also satisfying the access requirements of users of these materials. Traditional models of access, which focused on the service desk and reading room as means of mediating complex systems of indexing and identification of materials, fall apart in delivery contexts shaped by computer hardware and virtual libraries of born-digital materials. This is a problem not just for software history, but for every field of cultural inquiry. Literary drafts, correspondence, graphics media, data, and images created in the 1990s are more likely to reside on disk or in networks than on paper, and the trend, as an optimistic stockbroker might say, is upward.

This issue of access to digital documents and software strikes me as urgently requiring new institutional and curatorial models. Let us consider again the divergent roles of archives, libraries, and museums. W. Boyd Raymond argues in an article on how electronic information is reshaping the roles of these institutions that "the functional differentiation of libraries, museums and archives as reflected in different institutional practices, physical locations, and the specialist work of professional cadres of personnel is a relatively recent phenomenon. This functional differentiation was a response to the exigencies of managing different kinds of collections as these have grown in size and have had to respond to the needs and interests of an ever enlarging body of actual and prospective users." Raymond's view is that individual scholars continue to favor the ideal of a "personal cabinet of curiosities" finely tuned to specific research, an ideal that considers format of artifacts and media as irrelevant, while stressing content. This was the "undifferentiated past" that these institutions hold in common.[16]

The often synonymous usage of "Special Collections" and "Rare Books and Manuscripts" as designations of library programs will change as a result of collections of digital media and software. This will be a permanent change, and we cannot expect the traditional Special Collections community to come up with all the solutions for preserving texts and other cultural artifacts of the age of ubiquitous computing. One possible approach to solving problems will be a new functional consolidation of media collections, digital libraries, and software archives. The creation of such cabinets of media curiosities would assemble specialists in curatorial domains that are now separated, while cutting off the uncontrolled extension of established departments of special collections to digital materials and refocusing their attention on the venerated realms of rare books and manuscripts. Still, as Swade has noted in his writings on collecting software, it is tempting to lay aside theoretical problems of proper custody for software and worry instead about the work. The conundrum here is that while the relationship of software to hardware, its storage on physical media, or its association with artifacts such as disks, computers, and boxes, might lead one to think of software as fit for the museum, requirements of scholarly access such as identifying and locating sources, standards of indexing and meta-data creation, and maintenance of collections for retrieval and interpretation seem more in line with the capabilities and programs of libraries and archival repositories. In short, ad hoc decisions about curatorial responsibility may well have long-term implications for future scholarly work.

Kittler's admonition that "there is no software" provides little relief to archivists and librarians who have discovered that there is more of it than they can handle. And yet, the separation of physical media from content offers at least a glimmer of hope that the hard work of software history might be accomplished through a mixture of revised organizational models, new technological skills, and established practices, shaped by a re-convergence of museum, library, and archival curatorship.

---

[1] Friedrich Kittler, "There Is No Software," C-Theory: Theory, Technology, Culture, no. 32 (Oct, 18, 1995), URL: http://www.ctheory.com/article/a032.html.

[2] "The Computer for the 21st Century," *Scientific American* (1991). I am using the draft Weiser posted at URL: http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html.

[3] "Ubiquitous Computing," URL: http://www.ubiq.com/hypertext/weiser/UbiHome.html.

[4] Tim Lenoir, "All but War Is Simulation: The Military-Entertainment Complex," *Configurations* 8 (2000): 289.

[5] Luanne Johnson, "A View from the Sixties: How the Software Industry Began," *IEEE Annals of the History of Computing* 20, no. 1 (1998): 36-42, provides a summary of this development.

[6] Martin Campbell-Kelly, abstract of "Development and Structure of the International Software Industry, 1950-1990," Conference on "History of Software Engineering," Schloß Dagstuhl, Aug. 26-30, 1996, URL: http://www.dagstuhl.de/DATA/Reports/9635/campbell-kelly.html.

[7] Paul Edwards, *The Closed World: Computers and the Politics of Discourse in Cold War America* (Cambridge, MIT Press, 1996). Quoted from excerpt of chapter 1 at URL: http://www.si.umich.edu/~pne/cwpref.htm.

[8] Located at: http://www.links.net/vita/web/start/.

[9] For the diary entry, see the entry for Thursday, May 17, 2001: http://www.links.net/daze/01/05/. The eBay auction result is located at: http://cgi.ebay.com/aw-cgi/eBayISAPI.dll?ViewItem&item=590433613. For the article, see Justin Hall, "The Gaming Wars," *salon.com* (May 18, 2001) at: http://www.salon.com/tech/log/2001/05/18/e3_hall/index.html.

[10] This story is summarized thus on the "Old Man Murray" website at http://www.oldmanmurray.com/realnews.shtml: "As a measure of what deep financial trouble Salon is in, I respectfully - and bravely - present this ebay auction and the subsequent article on Salon. If you discount Wagner James Au's weird, finger-wagging, Amish-among-the-English story, Salon's only coverage of E3 was 900 words they won in a kid's ebay auction for $14.50." Quoted at "Print Links to justin/Links from the Underground," http://www.links.net/re/print/.

[11] Doron Swade, "Collecting Software: Preserving Information in an Object-Centred Culture," in: Seamus Ross and Edward Higgs, eds., *Electronic Information Resources and Historians: European Perspectives* (St. Katharinen: Scripta Mercaturae, 1993): 94.

[12] My talk was followed by Brewster Kahle's presentation on The Internet Archive, which is located at: http://www.archive.org/.

[13] James W. Cortada, "Preface," in: *Archives of Data-Processing History: A Guide to Major U.S. Collections* (New York: Greenwood, 1990): ix.

[14] From the version provided by the Software History Center at URL: http://www.softwarehistory.org/

[15] This brochure was later reproduced in the *IEEE Annals for the History of Computing* 2 (Jan. 1980). The text of this brochure is available via the website of the Software History Center at URL: http://www.softwarehistory.org/.

[16] W. Boyd Raymond, "Electronic Information and the Functional Integration of Libraries, Museums, and Archives," in: Seamus Ross and Edward Higgs, eds., *Electronic Information Resources and Historians: European Perspectives* (St. Katharinen: Scripta Mercaturae, 1993): 227-43, esp. 232.