III. EXTANT DOCUMENTATION & ITS SOURCES

A. POTENTIAL REPOSITORIES

To collect it, we must first establish where it is. Where is software documentation in the broad sense in which we are using the term?

To begin with, it is not primarily in the collections of other cultural repositories. The existence of other repositories interested in or already pursuing software archiving would be welcome; we know from the size of the universe that no one archive can be comprehensive. However, a telephone survey of archivists at a few computer companies, high-technology firms, universities, professional associations and governments which was conducted from March to May 1987 to determine whether they collected such material, and whether they thought the records of software still existed in the institutions for which they were responsible, exposed that they currently are not collecting this material.[17] Though incomplete, this survey exposed some salient facts about the enterprise.

As assumed, no one seems to have formed a software archive nor is anyone collecting software documentation, except as an inadvertent bi-product of institutional archives, with the noteworthy exception of one computer industry sponsored historical center in the United States and a similar effort just formed in the U.K.[18]. Even organizations such as SHARE and the Federal Software Exchange, which were formed for the purpose of "archiving" software in the data processing sense (that is keeping it for distribution during its useful life), have not retained their "back stock". A large number of such distribution programs still exist, however, and those which do (SERAPHIM, SIMTEL, EDUCOM, the University of Waterloo, SHARE, FSE etc.) would probably be very conducive to depositing their materials.[19]

---

[17] The survey, conducted by the author for the Computer Museum, was informal, but nevertheless revealing.

[18] The Charles Babbage Institute, 103 Walter Library, University of Minnesota, Mineapolis, MN 55455 and the National Computer Archive, Department of Science & Technology Policy, University of Manchester, Machester M13-9PL

[19] The Simtel 20 file server at the U.S. Army Base at White Sands, NM has a large library of public domain and user supported software which was originally built at MIT,

The survey discovered that the few repositories which are sophisticated in collecting machine readable materials, have focussed their attention exclusively on how to get the data out of software specific environment and formats in order to *dispose of the software*. The only exception to this practice to date has been in small pilot studies of DBMS and electronic mail environments where the question has been posed whether the documentation of such systems will require retention of the software.[20] Even in these experimental settings, however, the focus has been on how to avoid keeping the software (and being dependent upon it). Therefore we need to make a radical distinction between a machine readable data archive and the concept of a software archive or collection.

The concept of collecting software for historical research purposes had not occurred to the archivists surveyed; perhaps, in part, because no one ever asks for such documentation! For a variety of reasons best traced to the influence of social science data archivists over the development of machine readable archives guidelines, American archivists do not see software as having any evidential significance, and thus do not collect it as an aspect of documenting the ways in which their organizations worked.

While technological barriers have not been the reason why this material has not been collected, they would have become a reason immediately had the respondents given the matter any attention. The survey revealed that archivists assumed that executable software would be a necessary component of a software archive, if not its only holdings, and had ruled out collecting software documentation because they couldn't imagine how an archives could maintain facilities to run any software which might be acquired. When pressed to state precisely what kinds of research would require having the code and machines on which to execute it, no one was able to come up with a convincing illustration.

Once introduced to the concept of software archives, especially if such collections could be maintained without requiring the repository to become a

---

including libraries for a large variety of operating systems (especially UNIX, Ada and CP/M) and is available to users of Arpanet (or Bitnet which accesses Arpanet).

[20] John McDonald, "The New National Archives of Canada and Electronic Records, Archival Informatics Newsletter vol.1*2, Summer 1987, p.14-15

scientific endeavor[21]. The accepted literature on methods of archiving machine readable files, on archives automation and on the challenges of documenting database management and electronic mail environments, all view the software environment in which such new systems are implemented as a barrier to archives, rather than as a potential subject for documenting[22].

B. SOURCES

The process of developing, manufacturing, distributing and selling software is not fundamentally different from other research and development and marketing processes and can be expected to generate similar types of records except in a few respects, noted in more detail below. The best assessment of the significance and utility of different types of records, in the context of the place they played within the R&D process, is presented in the volume by Haas, Samuels & Simmons[23]. Unfortunately, that otherwise exemplary study provides no guidance regarding software, and little with respect to processes driven by software. When we consider that most of the new tools acquired by modern R&D laboratories are either software or software based integrated systems, the importance of providing better guidance is clear. In addition, archivists concerned exclusively with the preservation of data compiled by computers have made significant progress in the past decade in establishing standard methods for archiving of machine-readable data files. Their guidance asserts that the data should be rendered software independent by rewriting it to sequential data tapes documented by fixed field code books. This practice, while preserving the data for future social scientific analysis, has the potential of seriously impacting on our ability to understand the meaning of the data to those who

---

[21] Joan K. Hass, Helen W. Samuels & Barbara T. Simmons, Appraising the Records of Modern Science and Technology: A Guide (Cambridge, Mass., MIT, 1984)

[22] Hedstrom, Margaret L; Archives & Manuscripts: Machine-Readable Records, SAA Basic Manual Series (Chicago, SAA,1984). See also, Thomas E. Brown & William A. Reader "The Archival Management of Machine-readable Records from Database Management Systems: A Technical Leaflet", Archival Informatics Newsletter, v.1*1, Spring 1987 p.9-12

[23] op.cit., *21

operational facility for obsolete computing systems, there was considerable interest in the idea. Many of the interviewees requested copies of the final report and without prompting suggested that their institutions would be interested in pursuing the matter further. Most of the repositories contacted were, of course, natural components of a software archiving consortium or reporting network.

The absence of a community of scholars needs to be addressed as the mission of the software archive is articulated. The interest of other repositories in potentially collecting software needs to be exploited as a conscious element of the collecting strategy. And the distinction between the functions of the software archive and those of data archives needs to be made a clear focus of future presentations of the concept.

In discussing of the kinds of materials which exist in their institutions (as a postscript to explaining that there was no software archive), archivists identified numerous forms of material which would be invaluable as part of a software archive. This suggests first, that substantial documentation surrounding the history of software still exists, and second, that those responsible for it may not recognize it as forming a potential part of a software archive. However pleased we may be that some material has survived, both parts of the response represent a threat to the enterprise. Even archivists intuitively regard a software archive as consisting of code, and are therefore likely to discard other documentation. The AFIPS brochure "Preserving Computer Related Source Material" could help to raise awareness among creators of the documentation, but getting archivists to preserve appropriate source materials will depend on clearly articulating the mission of the software archive and actively promoting the concept. Advocates will need to plant the notion that the history of software is a broad-based research arena in appropriate places in archival literature. Even such sources as Appraising the Records of Modern Science and Technology: A Guide, though only two years old and written by potential participants in the software archive movement, barely refer to computer software and don't suggest the value of documenting it as subject rather than object of the

compiled it (ie. how they were able to use it in its native implementation). It is also easily misread by archivists without any technical training to recommend that they dispose of software and software related documentation, which could reduce the universe of such documentation.

Organizations which take part in collectively building the documentation of software should develop a shared strategy for defining the universe of documentation and determining collecting priorities. While such a strategy will be useful in building collections, it is not the same thing as the tactics for building the software archives. Rather it is a repertoire of tactics for analyzing the field of software history. It does not begin with records, and certainly not with surveying, nor is its principal intent to secure the deposit of records in a dedicated software archive, though it aims to assure that appropriate documentation does get accessioned by some repository, and that in the aggregate, this documentation and information already available from other sources adequately documents the history of software.

A documentation strategy begins with historical research which establishes what the significant features of an historical landscape are. Just what the problem domain includes, can vary from a specific intellectual feature (a documentation strategy for artificial intelligence, such as is dicussed below) to a general social effect (like the intelligent consumer product).

The initial questions are posed from two perspectives: what happened, and what types of documentation would have been generated by what happened. In the case of the intelligent consumer product, microprocessor chips capable of holding small instruction sets, such as the options available for a dishwasher or burglar alarm, became sufficientlly inexpensive and displays became sufficiently reliable at about the same time that they were associated somewhat romantically in the minds of consumers with high-technology which was perceived as valuable. The result was explosive. The first tactic for documenting what happened, and where, is to take a snapshot of the ephemeral consumer product literature and associated trade publications every two months for five years. What would be added by other forms of documentation? Perhaps the flavor. If so, interviews with industry executives would seem an appropriate second order documentation tactic.

The documentation strategy establishes what needs to be documented

and, in principle, how it might be documented, but part of its repertoire will necessarily include a survey, at least from reference sources, of existing records. The operating assumption here is that most concepts can be documented in a wide variety of different ways. There is rarely one single piece of information which is critical to understanding the past, and if there were, we could not know it was there before finding it anyway. Instead there is information which is known to a variety of actors, organizations, and associations and there are records which range form discarded audit trails and drafts of grant proposals to tax returns and annual reports.

1. Actors

After compiling lists of individuals who played significant roles in the area under study, the archive could assist them to place their professional papers in appropriate archival custody. To assist in the education of individuals contacted, other history of science centers have used a brochure for scientists originally developed by the American Institute of Physics to help to explain why and how to arrange for personal papers to be deposited in an archive. This brochure, or a similar one focussed on software documentation issues, would be a valuable tool for explaining to potential donors why their records are of interest and what they can do to assist.

Institutions are actors too, and while it may not be desirable to acquire their records in many cases, dedicated software archive programs, such as might be formed by the Computer Museum, the Charles Babbage Institute and others, could offer assistance to software houses, computer firms, universities and research centers, in developing corporate archives for their own records. These dedicated archives should avoid seeking to acquire such records for themselves, except as repositories of last resort, for reasons of self-preservation and politics. The records of a large software corporation or computer company are simply beyond the capacity of a Computer Museum, Babbage Institute, or even the Smithsonian Institution. The costs of entering into such arrangements, even with a very tiny firm of great importance, are that it is nearly impossible to ever back away from the relationship, even if the firm succeeds and grows beyond all expectations.

The records of professional organizations may prove more interesting than they sound. However, the disadvantages of corporate archival relations

in the for-profit sphere are magnified among non-profits. One significant exception is academia. One of the most promising areas of pay-off for the development of software archives is within university archives which could focus attention on software development and use in higher education. By documenting their own backyards university archivists could preserve much of the valuable record of software history.

A concerted effort to track down the records of the many informal and transient committees and groups, such as standards organizations and industry forums, which have played significant roles in the development of software is called for. There is no other way of acquiring this material besides systematically following all leads and tracing whether it exists. Models for such survey projects are available.[24]

2. Products

Searching for documentation surrounding a product can be a useful way to define the universe of documentation. Certain languages, specific packages, particular algorithms, have an influence on an area of computing at one time. These pieces of code and logic may have been developed by individuals whose work wouldn't otherwise be documented except that they were members of a team, or participants in a conference, or founders of a bulletin board, where the target product evolved. Often by beginning at the product end we can identify forms of documentary evidence and subjects of documentation which would otherwise escape us.

Software development does not differ greatly from other creative enterprises. It is esssentially an authoring process, whether done by an individual or a team. Obviously if a team is involved, more explicit outlines of the product, functional decompositions of its modules, and formal definitions of its interfaces, can be expected to have been produced prior to and during the process. Often systems written by individuals will have such documentation created for them only after the fact. For historical research, contemporaneous documentation is much more interesting, since it reveals changing

---

[24] Fleckner, John; "Archives & Manuscripts: Surveys", SAA Basic Manual Series, (Chicago, SAA, 1977)

assumptions, expectations and approaches. As with literary products, more can be made of drafts and discarded code than they deserve. Sometimes such abortive efforts and early stabs are interesting, but except in the case of the extraordinary product and exceptional creative genius, these materials will never be used by researchers and, indeed, little can be learned from the gropping progress towards a finished product.

Software makes generalized machines behave in specialized ways. But there are two ways to skin this cat; the other is to design the hardware to behave in the desired fashion. All software is, to some extent, making use of particular facilities of the machine environment (some of which may derive from lower level software) in which it is running. Ever since the very earliest machines, the engineers who construct the hardware and those who write the software have been separate; one of the most important and interesting issues in the history of computing is the interaction between them. It is truly a two way interaction with some functions migrating from hardware into software (typically to make them more flexible and adaptable) and others migrating from software into hardware (usually to optimize performance).

These adjustments generally take place in the development process, as part of the testing and tuning of systems, and over the course a system's life-cycle, with new releases of both hardware and software. Documentation is likely to be found in internal memoranda and progress reports; indeed without these, identifying the causes of changes in code from one iteration to the next would be difficult. Alpha and beta test documentation, will be explicit about the reasons for these changes (in the case of beta sites, the needs of users will be concretely stated here, if not in the design documents). The minutes and petitions of users groups, if these can be found, will also help explain both interactions and unilateral adjustments to hardware or software. Published reviews, benchmarks, post-bidding analysis in corporate offices which lose a major contract, and other critical assessments, also have an impact, and are excellent documentation to have in conjunction with the altered product.

3. Social and Intellectual Relations of Software

A large number of the subjects of the universe of documentation we are

seeking are not concrete entities at all, but patterns, networks of interpersonal relations or ideas. The record of influence can be every bit as physical as the object code of a system, but the subject itself is intangible. What kinds of strategies can be employed to expose documentation of amorphous things ? When seeking documentation of subjective states of mind, as we often are, it may be essential to resort to awards, rewards, and competitions to smoke out those individuals who identify with a particular belief, or participated in a debate. Gimmicks, like competitions, story telling contests, and recognition to donors can be used to attract items or classes of items to the archives. For example, in 1985-86, the Computer Museum conducted a very successful "earliest microcomputer contest", which resulted in its acquisition of evidence of this special class of developments and to a very well received exhibit[25]. It also rewrote the history of micro-computers by smoking out large numbers of individuals whose independent work was submitted for the contest but was previously unkown to historians.

The interaction between research on the historical structure of an area of potential documentary interest, and of surveys of documentation in the field (identifying the published materials, including histories, and the existing repositories of archival records), will create an further understanding among staff of a program which collects software of the nature of informational lacunae. As materials are identified for deposit in the software archives or elsewhere, the patchwork will be filled in and emphasis in the collecting endeavor will shift. Maintaining a strategic orientation, in which the objectives of "adequate documentation" have been articulated in advance and remain in place throughout the effort and against which particular collecting succeses and failures may be measured, is important both to the staff and to outside researchers. In the mid-1970's, Roy McCloed (then at the University of London) conducted a survey and documentation project on British science which reported both on where documentation was located, as well as what had been sought[26]. It even documented explicitly what documentation

---

25 The Computer Museum Report, vol. 17, Fall 1986 "The Catalog of Personal Computers" discusses the project and its reults.

26 MacLeod, Roy; Archives of British Men of Science, 1972

which had been sought had not been turned up in which places and what documentation was reported (and by whom) to have been destroyed. In retrospect it is clear that the negative reporting of the British Men of Science archival survey project was every bit as important to researchers as the positive reporting and that the combination of such an on-going account of documentation status and a clear sense of what constitutes adequate documentation is a mechanism for steering a documentations strategy and for focusing efforts where they are most needed.

More than most products of our society, software has been influenced by its own seedbed, the university teaching environment. A number of significant computing languages have their source in instructional needs (BASIC, LISP) and, because early computing was born in universities, so do many basic design concepts. Computing remains an industry which relies heavily on academics to forge new concepts. Immense programs are operated by each of the major computing manufacturers to give computers to universities and the Federal government is relying on super-computer centers associated with major research universities to give the United States the lead in the next generation of computers. Records of grants by industry and government to academics, and reports on their findings, will play an important role in documenting the history of software. Fortunately, many universities will be retaining much of this record as a routine part of their archival programs. Unfortunately, few of these programs were founded prior to 1970, and they have much to collect. Hopefully, alerting archivist to the value of these records for the history of software will increase their likelihood of retention.

To appreciate the social relations of software, we must understand what it means that software may be a commodity. While, ownership rights in software were until recently poorly protected by each of the methods available (copyright, patents, code hiding and trade secrets), any method of protecting ownership involves making and keeping records in the offices of the corporate counsel. Now that copyright and patent protection is becoming stronger, we can expect to see greater use of these methods, and greater historical interest in the files of patent attorneys and the copyright office. The requirements of the trade secrets law are such that those who use this as a method of protection have had to construct files demonstrating how they invented or developed a technique which gave them a competitive

advantage. These records, also found in corporate legal files, will be of future interest to historians.

Since software first emerged as a commodity it has been the subject of a massive ephemeral literature (in print and audio-visual formats) of advertisements, brochures, catalogs, demonstration disks, and endorsements. If the history of other aspects of our industrial society, such as the flowering of the machine age in the second half of the nineteenth century, are any indication, historians will find these ephemera (if they find them at all) to be extremely valuable evidence. Those of us who live with this stuff crossing our desks every day know that we are talking about immense volumes. Strategies for capturing this quantity of ephemeral literature need to be developed, whether they are cooperative collecting along divided lines of responsibility or random or periodic sampling.

Alexis de Tocqueville observed that American's organize for any and all purposes, and they still do. There are hundreds of organizations of persons involved in developing and using software. As formal mechanisms for communication and as part of the phenomenon themselves, documentation of these organizations will be critical to understanding the history of software. Some of these organizations, like the Institute of Electrical and Electronic Engineers (IEEE) for instance have already formed archives. Others have elected to deposit in existing archival repositories; the American Federation of Information Processing Societies (AFIPS), for instance, deposits its papers at the Charles Babbage Institute. Still others are exploring such arrangements; the American Society for Information Science (ASIS) is in negotiation with two potential repositories. Membership organization records are often thought to be barren, but committees of the Association for Computing Machinery (ACM), the American National Standards Institute (ANSI), the IEEE, and other organizations have played critical roles in the development of standards which shape software, and provided for training and professsional education. In addition they have testified before Congress, taken part in international conferences and trade delegations, and sponsored software exchanges among members and even the development of specialized software to meet shared needs.