

IV. THE MISSION OF A SOFTWARE ARCHIVE OR MUSEUM

A. DISTINCTIONS BETWEEN A SOFTWARE ARCHIVE, SOFTWARE LIBRARY & ARCHIVE OF SOFTWARE

The first step in establishing a software collection, assuming questions about its feasibility have been satisfactorily answered, is to define precisely what its purposes are, and are not. It may be useful, because we are defining a new kind of cultural institution, to contrast it to existing institutions before attempting an independent definition of its mission.

A software collection in an archive or museum differs from a software library in that its primary aim is to document the history of software or the manner in which the parent organization employed software, while that of the software library (even if, following data processing usage, it calls itself an "archive") is to provide software for the short term use of its clientele²⁷. As such the software archive supports research about software, not research using software. Obviously, the products which are acquired by a software library, for use by clientele of that institution for the purposes which the software was intended to serve, and which are likely to be among the most widely used and "popular" published software packages, are also part of the universe of a software archive.. The obverse, however, is not true: software documentation acquired by software archives, including documentation of developments which were not commercially packaged and of aspects of the environment of software other than its code, are not likely to be acquired by software libraries.

It is useful in addition to differentiate between two classes of software archives. A software archive may either be a distinct historical collection or a collection within an historical repository, or it may be an integrated component of an institutional archive, documenting how that institution used software, and thereby providing evidence of the functioning of the organization. In the latter case, evidential reasons may be the primary, or even sole, reason why an archival program retains software. Indeed, I argue strongly that corporate archives should retain software documentation for such evidential reasons.

²⁷ Robertson, Steven; Public Microcomputing: facilities and usage in public libraries. (Studio City, Ca; Pacific Information, 1986).

In contrast, the focus of the software archive as an historical collection is on exemplars of a creative genre, rather than on evidence of the activity of an organization or person. This kind of software archive resembles a literary, art or music collection, and like them it must collect not only specific examples of software but also materials which provide evidence of the way in which software was created, distributed and used. In this way it can support research on software itself, and on the contexts of software development and use. Just as we cannot understand art, without understanding the world of art patrons, art dealers and artists, we cannot understand software without documentation of its context of development, dissemination and use. The software archive will not consist of software products alone, without any documentation of process, and indeed might exist without any operational software code! A software archive can no more consist exclusively of code than a music collection could consist only of scores. It will necessarily hold a wide variety of forms of material including the correspondence of software developers, the business plans of sponsors, and the financial agreements of copyright agents. A software museum will also contain some examples of earlier computers and operating environments so that the software can be played to an audience which needs to know what it "felt like." Of course, the museum aspect of the software archive will also benefit from the range of forms of material, permitting exhibits to be developed around themes ranging from the sponsorship of software development to the nature of software advertising. The software library, most likely, would be interested only in holding the documentation required to "use" the software, for its original purpose. Therefore, the software archive will not be an archive of software, *per se*, but of documentation of the entire process of developing, implementing and exploiting software solutions.

Even though a software archive or museum will seek to further our understanding of the history of software through interpretative exhibitions and publication, it need not display software in an operating system to do this. Although exhibition purposes are best served when we can show as well as tell; an exhibit designer can teach much about software even if it cannot be demonstrated. Just as an historian of music can appreciate a score, without hearing it played, or especially without hearing it played on period instruments, a researcher in this field will be able to understand software and illustrate for others what it represents, without necessarily playing it

back on the machine for which it was written. Therefore, even for exhibit purposes, the availability of hardware and software environments appropriate for running a piece of software should not be considered a prerequisite for collecting it, although it should be taken into account in evaluating the exhibition value of the item.

In this respect, also, a software museum or software archive differs fundamentally from a software library, which has as its sole purpose the provision of software for use. Software which is not of use, would not be collected by libraries, nor should it be. But the software archive may acquire it, as well as isolated modules, routines or even sub-routines, not in full systems, since the novelty of the software, or its distinguishing features, may well be in a very minor component of the overall system. The archive or museum, therefore, may not be overly concerned about whether a compiler exists, or whether the source code itself can even be found, if other documentation makes it clear what the nature of the software innovation was, and supports a variety of types of scholarly queries. As a consequence of being machine independent, the software museum or archive can collect software developed in a wide variety of specialized contexts - such as navigation systems, energy management, robotic control, etc. While a software library must restrict itself to general purpose software, usually written of the kind of small, general purpose, computers it can afford to maintain for its users.

B. SCOPE OF THE SOFTWARE ARCHIVE

The archives of organizations in the software business, and of institutions which contract for or develop software to conduct of their business (most larger firms these days), will acquire a software archive by "retaining", rather than "collecting", materials. For them, the scope of the archive is not an issue, but for any repository involved in collecting software documentation other than as evidence of organizational activity, determining the scope of its collecting activity will be a major feature of the definition of its mission.

The decision to document the history of software in itself establishes a problematic goal. Unless we accept that no single archive, or small group of institutions, could achieve it, any more than they could document the history of government, we are certain to be frustrated. Organizations willing to collect software should institute a mechanism to share information with

each other about their respective holdings, and if, trust holds, about the contacts they are making²⁸. Competition is not pre-ordained. The interests of different institutions will skew some towards the financial impacts, some towards the use of software in a sphere of activity (such as communications, chemistry, education, or air and space R&D),²⁹ and some toward internal developments in software concepts. Any comprehensive long term documentation strategy will necessarily require numerous organizations with such distinctive perspectives to cooperate.

²⁸The Charles Babbage Institute has recently circulated such a documentation strategy for consideration according to Larry Hackman and Joan Warnow-Blewett, "The Documentation Strategy Process: A Model and Case Study", American Archivist, vol.50, #1, Winter 1987, p.12-47

²⁹These examples were selected because some collecting has already taken place at repositories which are naturally affiliated with communication (Bell Labs archive), automobiles (the Henry Ford archive), and aerospace (the National Air and Space Museum).