Software archives as a vital base for digital preservation strategies

Dirk von Suchodoletz
Albert-Ludwigs University
Hermann-Herder Str. 10,
79104 Freiburg i. B., Germany
dirk.von.suchodoletz@uni-freiburg.de

Klaus Rechert
Albert-Ludwigs University
Hermann-Herder Str. 10,
79104 Freiburg i. B., Germany
klaus.rechert@rz.uni-freiburg.de

Maurice van den Dobbelsteen
National Archives of the Netherlands
Prins Willem Alexanderhof 20
2595 BE Den Haag, The Netherlands
maurice.van.den.dobbelsteen@nationaalarchief.nl

***Abstract*—Software archives are vital parts in long-term preservation strategies of digital artifacts because almost every digital preservation strategy depends on additional software components. At this point a software archive becomes important. Such a repository should not only store ancient applications and special object dependencies like fonts or required helper tools, but also meta data like operation manuals, license keys and knowledge of operating ancient user environments. This article describes the challenges of establishing and managing software archives and presents a constructive approach which can be integrated in existing preservation frameworks. Furthermore, the suggested approach offers the possibility to interactively ingest software components into the archive and enrich it with sufficient meta data.**

I. Introduction

Memory institutions like libraries, museums and archives are already receiving large quantities of digital objects of various types, and the size and importance of this digital legacy will increase in the future. These objects could not be viewed or handled simply by themselves, but require a specific software and hardware environment to be accessed. Especially digital objects of many complex and specific formats are best handled by their matching applications. Independent of using a migration or emulation approach for preservation, just storing those objects themselves will not suffice. If e.g. a memory institution or company would like to convert their large quantities of Word Perfect 5.1 files to PDF they might face a significant challenge. If there is no simple tool for conversion for recent computer architectures available they depend on migration via emulation to use the original software. Conversion could be achieved by using the printing option to e.g. Postscript of the now historical operating system.

A memory institution receiving the legacy of an important writer, musician or politician is another scenario presuming a comprehensive and well managed software archive. Depending on the age, media and the popularity of the data format found, *digital-archeology* techniques requires access to the objects' original environment. The previous considerations are not only valid in relation to a long-term archives. It safekeeps for delayed and not yet preserved digital objects from different sources, which might be behind in long-term archiving.

Figure 1. German variant and a handbook of Word Perfect. It was a widely used text processing software from the end of 1980's up to the mid of 1990's. Several different versions were available for DOS and Windows operating systems.

II. Requirements towards object access

Regardless of whether using a migration or emulation preservation strategy, simply storing the objects themselves is not sufficient. If the representation of preserved digital objects has insufficient support on modern systems, the original environment has to be rebuilt, e.g. by using hardware emulation. However, the original installation media of the required software might be lost or might be inaccessible because of obsolete storage media or bit rot (Fig. 1). Furthermore, the objects to archive are not available in form of easy to transport bit streams, but are often still bound to their original medium.

As for most digital artifacts it is impossible to open a Microsoft Word or other text document on plain computer hardware. In general rendering or accessing an object requires a typically large and often complex set of software components: not only the original software application and an operating system, but also other dependencies such as font sets, decompression software, codecs for audio and video files, and hardware drivers for video output, sound cards and

peripheral devices. Hence, in addition to storing and managing the digital artifacts themselves, it is essential to store and manage a set of dependent software components too.

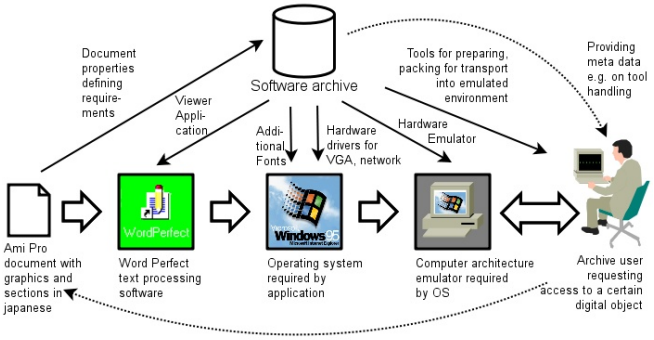

Figure 2. View paths formalize the pathway from the digital artifact into the archive users actual environment. They produce information on additionally needed software components to recreate a legacy software environment for emulation strategy.

Thus in addition to storing and managing the digital artifacts themselves, it is essential that we store and manage the required set of software components too. The dependencies between the originally archived objects and additional components can be formalized using view-paths (Fig. 2) for both emulation and migration approaches to preservation [1]. The number of dependent objects identified by a specific view-path and the objects significant properties varies. E.g. in some cases the process of rebuilding a specific environment requires the emulation of different architectures and operation systems because the emulator for the required system is not available for the host architecture. Therefore a software archive is required to preserve also operating systems, and the necessary tools.

An other important part of the software is the emulator archive of the different required computer architectures. A special subset of important software components are hardware drivers which were typically required to adapt operating systems to a defined hardware environment. We need to kept them for each virtual network card, video hardware sound or SCSI adaptor present in the collection of emulators. Simply using hardware emulation does not necessarily provide direct interfaces to the obsolete operating systems. But instead we have to re-implement certain once very common hardware components like the NE2000 network adaptor, the Sound Blaster 16 audio card and the Cirrus Logic VGA as QEMU[1] is doing.

For viewing or accessing the artifacts need to be passed into the emulated environment. This is typically a non trivial task and depends on the feature-set of the original environment. Depending on the computer architecture several types of floppy, optical or harddrives or network connections were available. Thus virtual container creation tools for accessing the virtual disks, floppies or optical storage of hardware emulators are required additionally for many tasks.

For the safekeeping of emulators, operating systems, applications and utilities are identical guidelines applicable like for the primary digital objects stored in the archive. Nevertheless it might be of interest to copy them to a directly accessible storage or prepare them for the convenient use in view path generation. Often requested view paths could be kept as combined caches of applications, operating systems and the emulator for faster access. Such specifically prepared containers could be distributed between the memory institutions to share the load of management overhead and costs. Beside

Figure 3. Without additional information and proper access rights to the object optimally stored with the metadata of the object you might stick with entry screen. Demonstrated is a rather ancient form of copy protection using a multi-layer paper disc distributed with Lucas Arts and other games.

the software components additional information and metadata like license keys, handbooks, operation manuals should be preserved and referenced in the software archive too (Fig. 3).

III. Legal concerns and practical aspects

Alongside managing the software components and associated documentation, a software archive must tackle the legal and technical problems of licensing. Copyrights are usually valid for a particular limited time period. Classical copyrights, depending on the national legal system, are applied over many years. They hereby outlive the widely intended life-cycle of an average digital object. This can mean that, if simply waiting for the end of any property rights, no technique or knowledge exists for handling the artifact anymore. A reputable institution

Figure 4. Selection of ancient installation media of popular software of the decade before last. The chances of copying the data from the carriers are decreasing because of obsoleted drives and carrier degeneration.

[1]A very popular Open Source X86, Sparc, PowerPC and other platforms hardware emulator, see *http://www.qemu.org*.

must abide by the licenses associated with the software it uses. For proprietary software, this may severely limit the rights of the institution to use the software to provide preservation services. Furthermore, technical approaches to protecting intellectual property, such as Digital Rights Management (DRM), copy protection mechanisms, online update or registration requirements all create significant problems for a software archive. Tackling this problem will require the cooperation of software manufacturers with a designated software archiving institution, to provide suitably licensed unprotected copies of software for long-term preservation purposes.

We recommend the development of an approach similar to the concept of the legal deposit approach used by many national or copyright libraries. Often the objects to archive are not yet in form of easy to transport bit streams in the archive, but are often still bound to their original medium (Fig. 4). This is especially true for the installation medium of all types of software: They are generally used for the setup of a machine. One of the main issues that has to be dealt with is the topic of possible legal implications of extracting and using instances of (old) software in completely different setups. From Emulation services could be offered over the network to archive users but this might be incompatible with the original licence terms.

Without licenses for proprietary software archive users are not allowed to use specific software to access or execute their digital artefacts. Beside the legal protection schemes other methods might interfere with proper software archiving: Depending on the computer platform, objects to be protected were made available on various data carriers. Protection measures on data carriers should prevent that the original installation medium can simply be digitally copied (Fig. 3) or the copy is worthless without a valid access or license key.

IV. Integration of software archives into preservation frameworks

To produce a valid and complete set of software components and describe them properly we suggest offering an interface to record a particular workflow once, such as installing a specific application to render e.g. a Word Perfect document and a printer driver to produce a PDF output from it [2]. Such a recording can serve as base for a later reproduction and in-depth problem analysis.

During the recording of the particular workflow the archivist should be supported by an interface to an online software archive (Fig. 5) for storing all additional necessary components like applications, operating systems, codecs, font-sets and hardware drivers for the emulated machine. This additional service would extend existing preservation frameworks like PLANETS [3] or CASPAR [4] to offer the possibility to interactively ingest this software into that archive and enrich it with sufficient meta data.

By using the view interface for installing applications and their dependencies all steps of the procedure can be recorded but also annotated by the archivist. For each installation step this information is kept together with the application files and

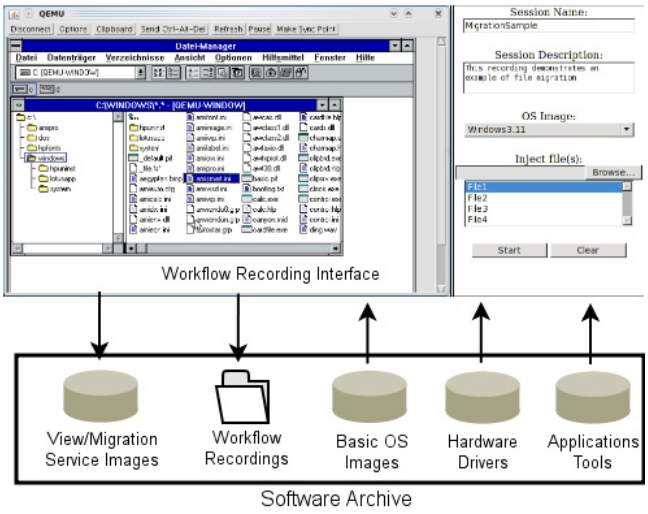

Figure 5. A user interface for archivists should be made available through preservation frameworks to the software archive.

the system setup, e.g. which transportation option was used to provide the installation image.

V. Software versions and variants

The complexity of software archiving increases if different variants of a digital object exists. Especially in interaction with users the localization of a software program becomes important. Humans depend on abstract user interfaces talking to them and taking commands somehow equivalent to natural languages. From a particular level of abstraction, localization becomes meaningful, which deals with the adaptation of the corresponding language. This occurs within the user interface – certain character sets of the keyboard input and the monitor output take care of switching to the users language. The operating system is typically responsible for this task.

While the user interface abstractly deals with localization, it must be concretely translated through the corresponding applications which are directly interacting with the user. This concerns the labeling of menus as well as all output in the form of hints, country or region specific formatting of currency, time and measurement units [5]. Beside localization different versions of applications (Fig. 1) play a role: The capabilities and file formats might have changed throughout the product cycle, e.g. modern MS Word versions could produce more complex documents compared to the early ones, but might not be able to interprete older files correctly any more. In some cases it might be useful to store variants of the same tool e.g. for different operating systems.

VI. Conclusion

Software archives are vital parts in longterm preservation strategies of digital artifacts. They are too important to solely rely on enthusiast efforts like of the Internet Archive or

abandon-ware initiatives.[2]

Legacy hardware drivers, older font-sets, codecs and handbooks for the various software will become more and more difficult to find. Experience shows that local and national solutions have not developed to the size of the task. Especially when dealing with localization a distributed approach among national memory institutions could help to equalize the workload for each partner. The new preservation frameworks should integrate interfaces to software archives to properly manage the requirements for stored digital artifacts (Fig. 6).

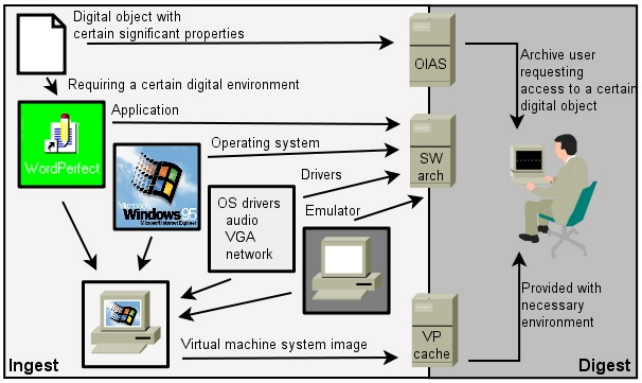

Figure 6. Future preservation frameworks need to include workflows for setting up of a software archive and the access to single components or aggregated virtual machine containers.

The suggested constructive approach, the software archive can easily integrated into existing preservation frameworks like PLANETS [6], [7] to offer the possibility to interactively ingest software components into the archive and enrich it with sufficient meta data. However, some important challenges remain unsolved. In general, digital rights management and long-term preservation have conflicting goals: Digital long-term archives should be enabled to copy and use obsoleted digital objects. Furthermore the access of objects for later users should not be locked to systems which at this point in time is with a high probability not available anymore. At this point legislative action is required.

An other specific challenge is non-standard scientific software used in rather special research setups, e.g. used in particle physics or astronomy to analyze the huge amount of data produced in these fields.

Software archives are a way that established archive organizations, like libraries or technical museums, could provide available tools for 'software archaeologists'. Despite the considerable efforts on digital preservation research, the essential groundwork on software archiving has until now been largely neglected. This could lead to fatal gaps in the preservation workflows of future generations.

[2]See sites like *http://www.softpres.org*, *http://www.abandonware.org*. Check Wired-Online article for some background information on the Internet archive initiative at *http://www.wired.com/culture/lifestyle/news/2003/10/60770*.

ACKNOWLEDGMENTS

Part of this work was supported by the European Union in the 6th Framework Program, IST, through the PLANETS project, contract 033789.

REFERENCES

[1] D. von Suchodoletz and J. van der Hoeven, "Emulation: From digital artefact to remotely rendered environments," in *iPRES 2008: Proceedings of the Fifth International Conference on Preservation of Digital Objects*. The British Library, St. Pancras, London: The British Library, 2008, pp. 93–98.

[2] K. Rechert and D. von Suchodoletz, "Tackling the problem of complex interaction processes in emulation and migration strategies," *ERCIM News*, no. 80, pp. 22–23, 2010. [Online]. Available: http://ercim-news.ercim.eu/images/stories/EN80/EN80-web.pdf

[3] R. King, "The planets interoperability framework," *ERCIM News*, no. 80, pp. 14–15, 2010. [Online]. Available: http://ercim-news.ercim.eu/images/stories/EN80/EN80-web.pdf

[4] CASPAR, "Caspar - cultural, artistic and scientific knowledge for preservation, access and retrieval," 2010. [Online]. Available: http://www.casparpreserves.eu

[5] D. von Suchodoletz, *Funktionale Langzeitarchivierung digitaler Objekte – Erfolgsbedingungen für den Einsatz von Emulationsstrategien*. Cuvillier Verlag Göttingen, 2009.

[6] PLANETS, "Planets - digital preservation research and technology," 2010. [Online]. Available: http://www.planets-project.eu

[7] OPF consortium, "Open planets foundation," 2010. [Online]. Available: http://www.openplanetsfoundation.org