Requirements towards Emulation as a Longterm Preservation Strategy

Dirk von Suchodoletz
University of Freiburg, Germany
Faculty of Engineering
dirk.von.suchodoletz@uni-freiburg.de

Abstract

Emulation is a strategy receiving increasing attention in the long-term digital archiving community. It can act as a complement to the otherwise dominant digital preservation strategies of migration and has convincing advantages, especially for dynamic digital objects. Nevertheless, a range of conditions must be met for successful reproduction of such digital objects in the future. The incorporation of view paths to identify the necessary metadata as well as additional software components may propose extensions of the OAIS reference model. This paper combines this view path concept, which captures the contextual information of software, with additional insights, thereby improving the flexibility of the approach. The concept provides insights into workflow instructions allowing for the archive management to preserve access to its artifacts. The view path model requires extensions of the metadata sets of the primary object as well as additionally stored secondary objects resulting in a reproduction of the object's environment, e.g. applications or operating systems. This paper addresses strategies regarding reference environments and gives an outlook for how to apply emulation strategies in the long-term while improving user convenience and maximizing emulation capabilities.

Introduction

The long-term availability of knowledge in digital representation presents mankind with a range of challenges. Organizations such as libraries or archives have for many years been confronted with the relatively new task of storing digital artifacts and making them accessible over long time periods. The composition of digital objects is a central feature of such objects, as opposed to traditional media such as paper or microfiche. These contexts change quickly and in order not to lose such information, they need to either be entrained with their technical development or their original work environment.

Migration is the method most deployed and trusted by memory institutions for long-term archiving of digital objects. This strategy takes digital objects through the constantly changing digital environment, made up of the changing hardware and software configurations. This usually requires alterations to the object and its inner structures. Although these adjustments make it possible to observe digital objects in the computer-aided environment of the archiver, this approach unnecessarily limits the amount of object types that can be archived since it is not applicable to dynamic and interactive digital objects. Additionally, problems arise from such forced changes, often making the authenticity of the object highly debatable, a problem which has become a central issue in the long-term digital archiving community (van Diessen and van der Werf-Davelaar 2002). The possible strate-

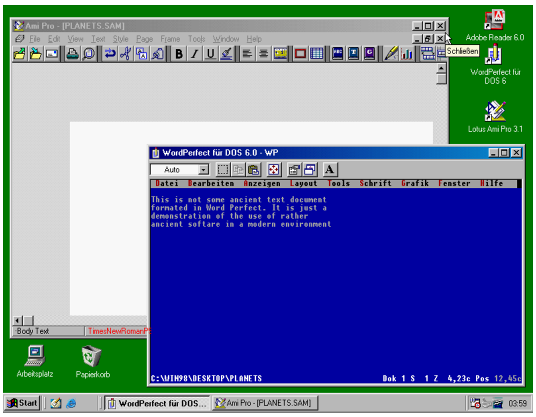

Figure 1: There is a rising number of dynamic, interactive digital objects which are not covered by migration preservation strategies.

gies for long-term functional archiving should be extended, so as to allow for access to the object in its original condition at any time. Emulation uses a different approach than migration, which is perceived by many as stronger. It does not operate on the object itself, but creates means to preserve its original digital environment. Emulation helps in becoming independent of further technological developments. This allows emulation strategies to avoid the alternation of the primary object of interest – the digital object being preserved.

Every layer of a software-hardware configuration can serve as a starting point for emulation. However, a number of key requirements must be considered, including the formalization of access processes, beginning with the object in question, and the actual working environment of the archive user. Pursuing emulation processes requires a whole array of considerations: the set-up of digital archives for a long-term archive, the type of working environment and the additional information needed for later access. This information could be gathered from the object's metadata, for example by using an already existing format registry. What

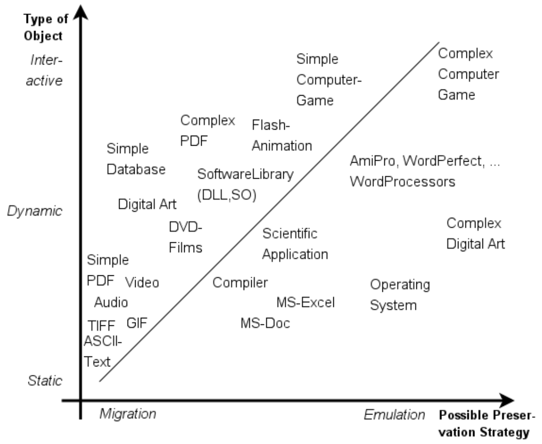

Figure 2: Suggestions for long-term preservation strategies according to different types of digital object types .

is generally true for digital objects is especially correct for emulation strategies, namely that digital objects cannot be accessed without technical equipment. Similar to the need for a record player in order to playback records, a tools archive is required for digital object preservation. Equally important for the reproduction of digital archives are appropriate reference environments and firmly defined software-hardware configurations. While the physical preservation of a technically fairly simple device such as a record player is not considered unproblematic, digital equipment is quite a challenge to preserve. This can be made simpler, however, by choosing the right method, avoiding the need to build and maintain a hardware museum. Instead, this task can be performed by a special software archive which contains hardware emulators.

The emulators' task is to bridge the technical past and future, as the software, like all digital objects, is affected by ageing. However, these emulators must be applicatively chosen and handled.

Static and Dynamic Digital Objects

With changing technologies, applications and communication, it becomes inevitable for the types of digital objects to change as well. With time new object types are added and others are discarded. Classical data types refer to digital objects in the form of virtual documents. Although these objects are no longer primarily recreated on paper, they are intended for potential reproduction on paper. This classical data class takes everything generated by authorities, businesses and organizations, reports, invoices or strategy papers into account. For most of this data it suffices to periodically assure its realization within a regulatory retention period. Digital artefacts of social or historical relevance should of course outlive such periods.

A feature characteristic of the above-mentioned objects is their ability to be reproduced, at least theoretically, at any time on a non-digital medium like paper or microfiche. Audio and video can be dubbed onto an analogue medium. This becomes more complex with, for example, the construction plans of long-lasting producer durable goods. Although they can be printed on paper, some information levels and associations are lost.

The above mentioned "traditional mediums" require minimal or no technical effort in order for end users to gather and interpret the content. This is no longer true for dynamic data and interactive objects. Up until the end of this decade, the long-term digital archiving community has given little attention to this problem. These objects deal with executable programs, components or software libraries. Included are: interactive mediums and educational software, databases, operating systems or applications. These dynamic objects are characterized so that they cannot be meaningfully interpreted and used outside a certain digital context.

With the widespread implementation of computers and the accompanying revolution in the data networks field, new areas in which digital objects are created have been added. While in the beginning stages of the internet, the World Wide Web (WWW) was still dominated by static content which had a lot in common with the aformentioned type of documents and images, this medium has clearly become more interactive over time.

Dynamic web content, supported by multimedia formats, like Adobe Flash, Java Script, applets and databases in the backend, dominates the large sites in the Internet. Many web applications carry out the bulk of their interactions on the server of the respective content provider, usually operating in conjunction with special clients. This field is being strongly extended, especially in the university domain, through the use of this medium as an education platform. The simple video recording of e-learning sessions and systems proves to be an unsatisfactory type of preservation in most cases. From interactive applications with a number of possible navigational paths, such a linear representation becomes limited in complexity and validity. For many national libraries, the topic of web archiving gains importance with newly added rules and legislation in place.

Databases were a very early field in the deployment of digital machinery. The moving, searching and linking of large amounts of data is a main strength of computers and has become a primary field of deployment. These electronic data collections often set the basis for derived objects. Different classes of database-based users include planning and accounting systems, like SAP applications, electronic timetables of miscellaneous public transport companies to Content Management Systems (CMS) of modern internet presences. Previously archived timetables of rail companies are a good example of large databases now yielding point new challenges. For example, as of 2008 the British Rail does not print timetables any more and publishes them as a printed book, an often requested artifact in the British Library.

Digital Archives as a Highly Demanding Task

Institutions such as libraries and archives have only limited influence on the generated and delivered formats. They are required to permantly maintain most digital objects readily available for access to users. For the less technically-experienced users, they must provide a suitable platform which allows the reproduction in common object formats. Since many objects can not currently be migrated to an appropriate form, this platform will need to support a large amount of older formats.

Digital archives pose new challenges to their curators. This goes along with the adoption of standards like the Open Archival Information System (OAIS). This standard, applied by the ISO, deals with the ingestion, management and digestion of digital objects. In its present formulation, OAIS offers only a rough specification of assembly and organization. For this reason current research projects and initiatives concerned themselves with its refinement and concretion. OAIS is not the only standard that deserves attention. In addition to digital objects standards, i.e. for data formats, metadata have standards for classification and meaning description. For digital preservation, the ideal would be to have all standards openly applied and free from proprietary components. This

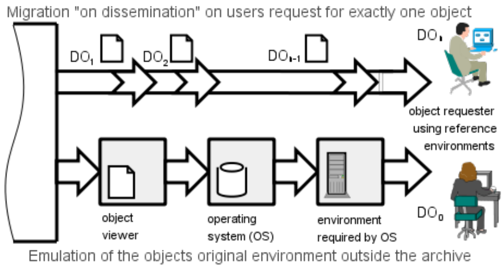

Figure 3: Independent of migration or emulation strategy, certain steps have to be implemented to actually render the digital object.

eases the task of archiving in the future, which is often adopted with a considerable delay and does not work backwards into the past. Furthermore, not all objects and process standards allow for themselves to be defined or established. Independence from actual object types occurs with the set-up of an OAIS-conforming archive enhancied with additional information, i.e the metadata. By now a whole array of metadata schemas have been defined, including Dublin Core (Gabriel and Ribeiro 2001) or PREMIS (LoC 2008). Such flexible and extendible models stand out and could assert themselves as a dominant standard within the field of long-term digital archiving.

Field of Controversy: Migration versus Emulation

While digitalization and standardization mark the front end of archive recording, migration and emulation serve to access digital objects. The way the strategies work can be perceived as conflicting. Migration carries an object along, including its technical advancements and keeps it readily available for the respective current computer-aided working environments. On the other hand, emulation positions itself in the environment of the respective object. Migration changes the object itself with all the advantages and disadvantages; emulation, however, does not. The efforts exerted for the management of a long-term archiving occurs, depending on the strategy, at different moments. Migration does have many disadvantages that prevent use with certain object types. However emulation also depends on a number of preconditions.

Migration

In the context of computers, migration is understood as the transformation of a digital object derived under perpetuation of an array of settings and under the use of certain criteria. This typically deals with the conversion of files into its respective representation, considering the current program availability. This procedure is often referred to as data format conversion, and should secure the optimal access of an object both currently and into the future. The concept of migration of computer files is not limited to long-term digital archiving. Alone the many data formats, resulting from the large amount of applications that can be created at any given moment, requires the permanent use of migration in order to swap data among one another. For this reason, many daily computer users are familiar with migration or conversion. Migration as

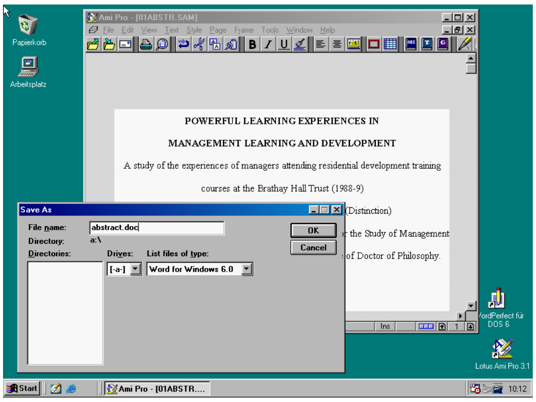

Figure 4: Demonstrated here is a typical vertical migration step using the application the object was created with. Saving the file in other formats offers users the possibility to import into more recent text processing applications.

a format conversion can be done either horizontally or vertically for a given time period (Fig. 5). Horizontal migration is the retracing of regular updates on the part of the software producer. Since every new version adds new functionality, this has inevitable effects on the data formats in order to represent the new settings. If older files created with the previous version are to be opened, an import function is required to change over to the new software.

Users of different software products for alike tasks are confronted with similar problems, but rarely is only one product available. Applications with similar functionalites are classified according to their intended use, whereby products can differ considerably in their functionality and operating models. One example would be the different applications for text creation, like Microsoft Word, StarOffice Writer or historical ambassadors such as Ami Pro, Wordstar and WordPerfect. Import or export between these applications is labelled as vertical migration.

It should be mentioned that no single migration strategy exists. Instead, migration is understood as a general concept containing a range of strategies. Especially the diversity of the involved data types requires variety in the handling of different files. Migration demands awareness of the choice of data formats and standards, the transformation between data formats, and the choice of migration between different physical media carrying data. One should

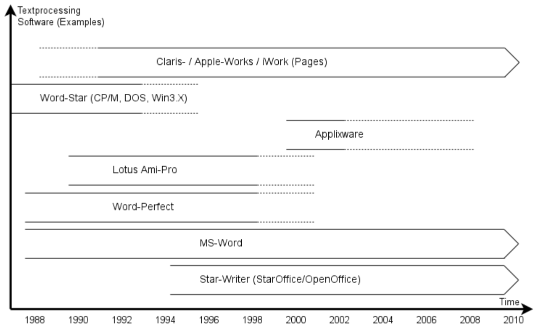

Figure 5: The domain of text processing software is an example of the typical migration challenges. Updating or changing the application documents were produced with most certainly requires format migration.

generally retain the migration process file transformation contents due to the possibility of data loss. If multiple transformation processes occur in succession, information losses can multiply (Fig. 6). A noteworthy problem is the authenticity of the digital object after the transformation of the logical structure. This issue is raised with nearly all migration processes such as the upgrade within a software product, the change of one vendor to another or the subsequent end of an application. The following challenges need to be considered:

- With every migration step, it should be ensured that the process, when reversed, delivers an identical output object.
- In addition, it is generally advisable that all output files, even those in intermediate steps, must be retained to allow for comparisons later.
- The first points determine the availability of the involved applications in advance. However, the problem of proclaiming a software, the failure of an organization or the constant upgrading may lead to doubts.
- Stepwise migration should be performed with all objects. Many software products require exactly this, since their "format history" implemented through import filters often doesn't go back very far. This approach is costly, however. A control should be performed on the quality of migration results which isn't automated in all aspects. The question of authenticity check affordability must be raised, since in most cases the mere inspection of a surface comparison is not sufficient. Furthermore, the data accumulates with every migration step, since new objects are constantly being created.
- Lossy compression techniques depart in relation to the preservation of authenticity. It could be

alternatively considered, to create special object copies, which allow bandwidth saving transport to an end user.

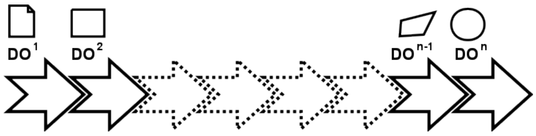

Figure 6: Authenticity of digital objects might be lost under repeated migration steps. Automated integrity checking is not easy to achieve but might be run using the strategies of (Becker et al. 2008) for supported formats.

Given the continuously growing amounts of data, a reduction of the total data volume may seem advisable. Sorting out apparently obsolete digital objects should be avoided, however, since predicting which object may later be needed is again a very difficult task. Furthermore, the main points of interest of later generations could change completely. This may also be difficult to predict.

The cost of transformations should not be underestimated. Regardless of the current predication of the relevance of a particular object, the complete data collection should always be migrated up to the required time point. Here, the accompanying measures of quality assurance and control should not be neglected, otherwise data loss may result. Migration and verification processes cannot claim to be irrelevant to computing capacities, since large numbers of objects will stretch over long spaces of time. Planning such a process was a main issue in the Kopal project.

A general problem is the continuous change of information technologies. This change is subject to the different software products and manufacturers, making the prediction of long-term durability of certain products or product groups difficult to get right. For almost all digital objects that have not applied comprehensive standards, the danger looms that the chain of possible migration steps will break at some point. At the end of the lifecycle the most important properties might be lost.

Emulation

Through the character found mostly in libraries, digital objects dominate migration. It attempts to "carry" the objects along with its technical developments, therefore making them accessible in working environments. This approach has many weaknesses and completely ignores dynamic digital objects. Another often unsatisfactorily answered question is that of migrator authenticity, resulting in the inevitable change of objects over time. Similar to old original factory expenditure or with legal documents, the future verifiable authenticity of digital objects is of great importance.

Emulation does not operate directly on the object but rather addresses the environment which was used to create the object. This means, for example, the replication of software through other software; in the best case, it won't make any difference whether it is handled through an emulated or original environment. One of the first to consider this was J. Rothenberg (Rothenberg 2002).

Emulators preserve or alternatively replicate old working environments. Thereby they achieve the connection of the current computing platform or a bridge to the creation moment of the object. Emulation first creates a virtual environment within a given working environment. This could mean software is cloned through another software, along with the hardware in software. Objects created in this working environment should largely run or look like the original platform or application. Emulators create the bridge between the current technology and outdated technology. Therefore, emulators must deal with the appropriate translation of the input and output.

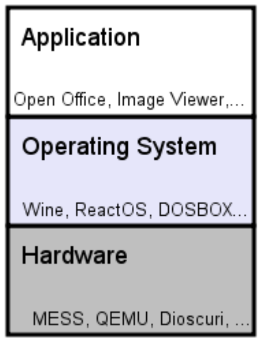

Figure 7: Generally, three levels of emulator implementation for current computer platforms can be determined: Topmost, the application layer followed by the operating system layer and on the lowest level the hardware layer.

Emulation or replacement of applications with others to execute or display particular data formats. For many operators, one of the top priorities is the emulation of features of old applications in currently implemented programs. The interpretation of other application current data formats is also a priority. Data exchange between the programs from different manufacturers is even possible. From the view of long-term archiving it is usually sufficient for the observer to work with a given object within the current working environment. Here it should be noted that the complete functionality isn't necessarily needed for the rendering and observation of an object. For an entire array of generic formats such as ASCII, Image or audio files, a number of viewers for the various available platforms also exists. The more specific or complicated the data formats become, the more unlikely the existence of an appropriate viewer which can reproduce the object properties in the desired quality.

Emulation of operating systems used to execute applications on an operating system that expects the interface (API) of a different operating system. The emulation of operating systems' APIs seems to be a general approach for the replication of applications. Theoretically, very old software can also be installed on modern operating systems as long as all called functions are in the form expected by the application. This cannot be realised with incomplete replication of required APIs. Since in the field of commercial propriety software many interfaces are not or only partly open, the probability of incompatibility is large.

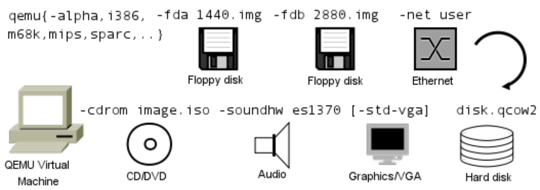

Figure 8: QEMU is a very popular powerful Open Source modular hardware emulator configured through command line for a set of different architectures like e.g. i386, Sparc and MIPS.

Replication of a complete computer architecture: Different from the programming interfaces of an operating system, the functionality of single applications or the description of data formats are open for most hardware platforms. No application or operating system must be written down or translated in order to access many thousands of digital objects. The function set of a hardware platform is straightforward and often much smaller than that of an operating system or application. There also exist much less hardware platforms than operating systems.

Emulators – Bridges between Digital Past and Future

Hardware emulators may play an important role in the long-term archiving of digital objects. If you would like to extend the class of accessible objects to dynamic, there is no other way to preserve the suitable working environment. As already mentioned, a hardware museum doesn't have a really reliable long-term perspective, since the archiving of old computers is for many reasons costly and risky. Therefore, a virtual collection of past and soon to be replaced computer architectures is a sensible alternative that has many advantages.

Emulators for different computer systems become a central element for a working software archive. This software archive may be composed of a substantial part of an OAIS based long-term archive. Emulators are also, like the actual primary object of interest, affected by maintenance, and share similar problems. The actual creation of relevant selection

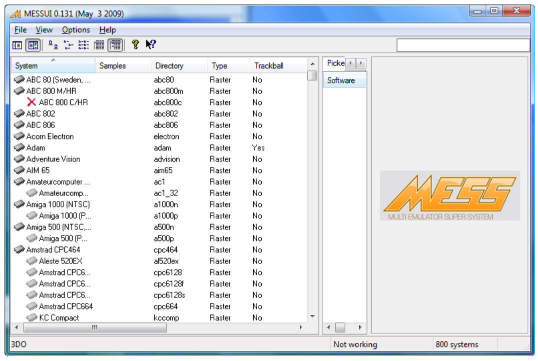

Figure 9: MESS is a modular emulator for homecomputers, consoles and ancient systems of the early era. It can recreate the hardware of nearly 800 different plattforms and derivates in software and could be executed on modern Windows, Mac OS and Linux desktop operating systems.

criteria depends on the respective archiving goals of the archiving organization. Metadata, the information delivered to an emulator or project, plays an important role for categorization and later observation. Included are:

- the service to deem classes of digital objects as important.
- the institutional framework and the identification of relevant user groups and their (predicted) requirements.
- the technical and non-technical characterisation and the estimate of the integrity of the replication.
- the indication of economic assessment and the estimation of long-term orientation.

Selection Criteria and Metadata

The appropriate selection of tools for long-term archiving is a fundamental success factor of the emulation strategy. Although it can be difficult to generate a conclusive list of selection criteria, since this depends very much on the user group, the type of primary objects and the planned presentation are important to the end user. On the one hand, a large selection of criteria increases the probability of the correct preservation of a working environment, but on the other hand, the cost of operating the archive rises with more tools.

Metadata and the selection criteria of emulators are closely related. Metadata is information which has been preserved for most digital objects in one way or another and must be preserved for the setup of a software archive. There is not a great difference between metadata and primary objects, however metadata provides additional information for intended use along side classical metadata. This extends the list of base information for a group of specific selection criteria. These selection criteria could result in a more complex matrix. Here, the high rating of a particular criterion makes the fulfilment of other criteria more difficult. Although it should not result in ”the one archive strategy”, you would favour a wider selection of tools. This allows for the achievement of ”view paths”, which are introduced in a later section.

The list of selection criteria can be categorized in groups:

- Generic metadata- like name, manufacturer or project, homepage in the Internet and history of changes, dependencies on the host platform, the software-hardware environment in which the emulator is being run and in its concrete execution- is also labelled as reference workstation.
- User groups, such as universities, research institutions, state or national libraries, have different requirements for object archiving periods and regarding the display of single objects, features in handling objects, and frequency of access on objects in their long-term archives.
- Technical criteria decide the features and application area of emulators, indicating performance, controllability and the eventual possibility of non-interactive use. Just as relevant are the requirements of the emulators for their technical environments. For long-term assessment and use, questions such as programming language and documentation play an important role. The technical criteria differ depending on the implemented environment. The hardware parameters of the emulation of an Amiga 500 are clearly more limited than the large variance in X86 PCs.
- Non-technical criteria are comprised of the economic assessment of a tool, such as licence cost and support. This also includes the product or project history, the market penetration, and the existence and knowledge of an application-community.

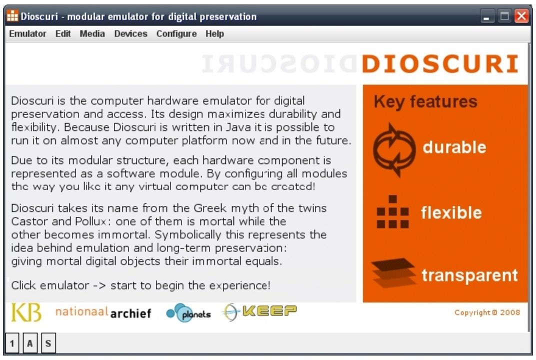

Figure 10: Dioscuri is a special modular emulator written in the Java programming language for the purpose of long-term preservation of the X86 architecture.

While until now no emulator has been programmed primarily for the purpose of long-term archiving of digital objects, this is changing with current research. Along with the already established emulators from previous times, Dioscuri (Fig. 10) a X86-Emulator is added which is being developed primarily in respect to long-term archiving. It however does not yet have the functionality range of other X86 emulators like QEMU (Fig. 8). Of great importance is the group of virtualization programs of the X86 architecture, at least for today’s activities. Typically implemented for entirely other reasons, they are suited for a medium-term perspective and considerably advance the idea and knowledge of hardware emulation.

Long-term availability of Emulators

Emulators themselves are nothing more than dynamic digital objects. The problems they face are the same as those for primary objects, which an archive user would like to observe. For this reason, the considerations for the perpetuation of the emulator for future use is a central component. Hence emulators need to be regularly adapted to current hardware and operating system combinations.

The possibility of software migration appears only with contradiction to the considerations undertaken initially. That is how the migration of dynamic digital objects, due to their high complexity or the impossibility of implementation on certain objects, was classified as inappropriate for long-term archiving. This problem is reduced by choosing a suitable emulator. If the emulator is available as an open-sourced package, it can be ensured that a timely adaption to the new computer’s respective platform occurs. Often it is about the integration of the current user’s platform interface. Durable programming languages such as C should allow for a translation with the respective current compiler. Both give certain stability to the paradigm, host operating systems, and user interfaces as well as the programming language. The advantage of this application lies in the use of only one emulation layer. This means that the interaction and the data exchange with the goal system can follow directly. If there is no possibility to port the emula-

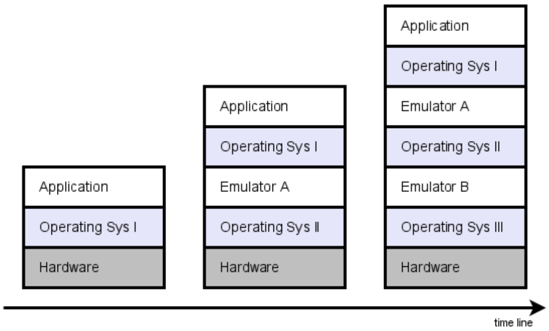

Figure 11: If it is not possible to update the emulator, its original environment might be emulated instead. Emulator stacking imposes challenges for object transport into its innermost environment (Verdegem and van der Hoeven 2006).

tor to a new host platform, the now outdated host platform, for which the emulator was created, can itself be emulated (Fig. 11). This is referred to as nested emulation. This is a considerable advantage in the avoidance of migration complexity. In the course of research in digital long-term archiving, approaches like UVC or modular emulation are being investigated (Lorie 2002), (van der Hoeven, van Diessen, and van der Meer 2005).

Examples of Available Emulators

Emulators for various computer systems therefore become a central component of a working software archive. Again, this software archive can be composed of a component of an OAIS-based long-term archive. The use of emulators is not new. Over the last 30 years, many have emerged, and here is a showcase of three for two hardware classes – the X86-PC and the currenlty unavailable class, the home computer – will be briefly introduced.

At the current time undoubtedly the most interesting emulator is QEMU (Fig. 8, (Bellard 2008)). Meanwhile, it has gathered a large user and developer community. QEMU was built primarily as an emulator of various computer architectures. The project is Open Source and is governed by the GPL. The source files are therefore opernly available. QEMU is an application without a graphical user interface. It provides an array of command line options for initial configuration and a monitor interface. The latter can be used to embed the run time of the emulator's removable drives. This way various types of control can be implemented. This includes the externally triggered swap of disks or the "insertion" of virtual optical media like the connection to USB devices.

Additionally, the keyboard input and mouse actions can be triggered through the monitor interface. Certain key combinations can be passed which are typically caught by the reference environment. This allows for entire program sequences to be automated, which would otherwise require direct user communication.

The Dutch national library, the national archive as well as the technology consultant Tessella are in cooperation developing a 16 bit version of an X86 emulator in Java – Dioscuri (Fig. 10, (van der Hoeven 2007), (DIOSCURI 2010)). The name of the project relates to the Greek legend of the twins Castor and Pollux of which one was mortal and the other immortal. This is meant to express the character of the emulation: "Finite" digital objects should put aside their immortal equivalence.

Java offers as an independent programming platform with the language necessary to create programs that can be run in different reference environments. Such applications avoid dependencies on certain combinations of computer architecture and operating systems. The emulator was applied as a modular tool, every hardware component exists as a standalone module and by combining components, a complete hardware emulator is generated, considering a certain basis set.

The vast majority of home computers are based on the processors of the 68000 series from Motorola or the Zilog Z80 Processors. To this basis, components like input and output building blocks are added, which do not significantly differ between machines. Therefore MESS (Fig. 9) follows a modular approach, which attaches a universal basis machine to each system, whether it be a games console, calculation, handheld or office computer, along with a description. Based on this, it calls back pre-made emulation code blocks.

The acronym MESS stands for Multiple Emulator Super System. MESS is an Open Source project and is governed by the GPL Licence. It is supported by a large developer community and is actively maintained. Windows variants are the predominant reference platform, followed by less supported systems like Linux or Mac-OS X. Since the firmware or operating systems of the individual devices are often under propriety licences, they are not a component of the emulator. They must be obtained, installed and suitably stored in a software archive separately.

From Object to its Representation

Digital objects cannot be used by themselves, but require a suitable context for their working environments in order to be accessed. This context, called the working or utilization environment, must combine suitable hardware and software components so that its creation environment or a suitable equivalent is generated, depending on the type of the primary object. View paths are therefore a central concept that is discussed later.

At any given time point, a primary object is requested by an archive user. In any case, the requested object requires a context which allows the requester to interpret the object. With an increasing time gap from the creation of the object, the probability that this context is available in the user's digital working environment is lowered.

Thereby the central task of the archive operator is determined. The task does not end with bit stream preservation, successful preservation and updating of archive objects, instead with ensuring access to all objects in the archive at any point in time.

For this, a suitable environment is expected, in which an archive user can observe or execute the object. The user groups of individual digital archives and collections vary in their degree of knowhow. Since an average user cannot necessarily be assumed to have extensive experience with computers, one must consider the proper accessibility and use of the required software and their interfaces. The operational procedures for the access of different object types can vary:

Migration-on-Dissemination. The primary object is – by type in multiple consecutive steps – converted to a form observable by the user. This process can involve emulation processes in order to run old migration software. It can, depending on the depth and complexity of the individual steps, take a longer amount of time, and under certain circumstances require direct user interaction. Migration-on-dissemination is suited only for static objects. The goal of migration is a format which can be displayed or played back on the viewer, available in the respective reference environment. In this procedure, the object is inevitably changed and may not meet particular requirements under certain circumstances.

Reproduction of a utilization environment. For all other mentioned object types, only the reconstruction of the original environment or an equivalent environment as a reproduction option remains. These processes is again decided by the concrete object type and its specific requirements. The processes involved will in most cases require a certain user interaction in addition to a certain number of secondary objects. Furthermore, the question of suitable primary object transportion to the restored working environment is posed. The

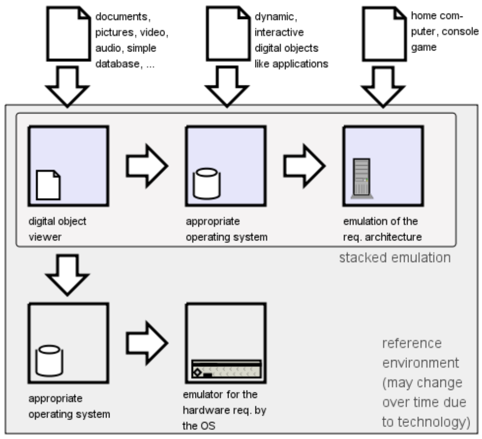

Figure 12: View paths define ways for a complete software (and virtual hardware) setup starting from the object, covering the rendering application, the operating system required by the object and the application and the emulator needed to bridge into a current desktop environment.

concrete presentation of the respective access depends on the archive operator and their guidelines as well as the user group. This can be done onsite, for example, through a reference workstation in the library rooms or through suitable transport over communication networks.

Digital Archaeology

The proposed considerations are not only valid in relation to long-term archives. It may occur that from different sources, which may be slacking, digital objects appear to not have yet been saved in a long-term archive. On such proposed processes, object type determination and display for the user can also be used. This way, established archive organizations, like libraries or technical museums, could make the tools available for ”software archaeologists”. Furthermore, it may be necessary to provide suitable reference hardware in order to read old data mediums and ways for importing external primary objects.

However, the described possibilities should not be regarded as reliable alternatives of a trusted long-term archive. While archive management for every incorporated object type can prove to be observable at anytime, this cannot by nature be done for externally delivered objects. The uncertainty factor of whether the archive organization is actually in the position to suitably deal with the delivered object remains.

Formalizing Representation Methods

The reproduction of utilization environments or corresponding equivalents is best illustrated and formalised by so called "view paths", sometimes named pathways. These are ways to get to the working environment of the observer or operator from the primary object of interest . The illustration of various view paths (Fig. 12) show a typical vector originating from the primary object over its creation application to the required operating system up to the resultant hardware emulator. However, regardless of the emulator chosen, contextual information of the computer environment is always required. For example, questions such as "for which operating systems is Wordperfect 5.1 compatible?" are less obvious today than they were twenty years ago. To overcome such knowledge gaps, a formalization process is needed to compute the actual needs of an authentic rendering environment of the digital artefact. In 2003, IBM Netherlands originally proposed the concept of a view path, based on their preservation layer model (PLM, (Oltmans, van Diessen, and van Wijngaarden 2004)) which was refined during the research on emulation at Freiburg University (van Diessen and Steenbakkers 2002), (von Suchodoletz and van der Hoeven 2008), (von Suchodoletz 2009).

While the origin of the view path is fixed to the primary object, the end-point of the view path adjusts, externally determined through technological advancements and successive obsolescence of available computer platforms. Furthermore, the length of a view path depends on the type of the primary object. Generally, the following scenarios arise for view paths:

- At any give point in time there is a way from the primary object to it observation or execution.
- There exist many different view paths (Fig. 13). These have suitable metrics for error, allowing an automatic decision to be made.
- Primary objects exist, that after a certain point in time, no view path can be constructed.

For meaningful determination of the existence of view paths, they should correspond to certain reference environments with respective fixed hardware and software. A view path can be interpreted as a decision tree with the primary object at its root. A leaf without branches composes the end of a path the in form of a valid reference environment. The assembly of the required working environment may be seen as a layered model, as shown in the figure (Fig. 7).

Having an emulator and the contextual information contained in a view path still leaves some implications before the rendering of the digital object. First, the digital object must be characterised which may be difficult if the object is not a single file but a group of files forming a program. Then, depending on the emulation level (Fig. 7), a number of additional software (secondary) objects like applications, operating systems, helper programs and drivers must be taken into account. Different view paths exist for each object type, increasing with the number of object types.

Many objects can be represented through more than one application (Viewer). This has different results in authenticity, complexity and quality. This yields a path branching and a choice on the layer of the application. Similarly affected is the requirement of an application for an operating system, so that in this layer a new branching can occur. The recursion continues with the operating system and a possible selection of suitable hardware emulators. Since reference environments are mostly externally given and can only be influenced by the archive operator, view paths and reference environments decide one another. Furthermore, operating systems and emulators over hardware drivers depend on each other.

The modelling of the view path in layers is not tightly fixed. For example, the number of layers in a digital primary object in the form of a program is reduced. Something similar is true for earlier platforms like home computers, whose "operating systems" were fixed in the ROM. Furthermore, layers can then again be stacked, if it is necessary for a particular emulator to create a suitable working environment, which is illustrated in the right part of the layer model.(Fig. ???)

Metrics and Cost Estimation

A meaningful extension of the somewhat fixed approach of the original DIAS Preservation model could emphasize the single view path option, which could be illustrated through a descriptive metric. When there is more than one option at a node, it would be sensible to:

- Allow user preferences, for example in the form of the choice of application, the operating system or the reference platform,
- Take weightings of whether, for example, a particular value projects onto a true to original, authentic representation or a particularly basic use,

- Allow comparisons between different ways to better secure the security and quality of the representation of the primary object,
- Estimate the cost related to the various view paths and where available, allow an additional economic evaluation.

A conclusion could be multiple dimensional metrics, which together with the object metadata are discarded and regularly updated through the archive management. This should occur with the feedback of the user, to allow for the stream of current valuations on the user side, which depend on the practical deployment of certain view paths along with their manageability and completeness.

An interesting problem, not so much from the perspective of the primary projects as from the archive user, becomes apparent from the various localizations. This refers to the adaption of the software to certain language areas and measurement units. This includes the various currencies and their symbols, measures or the format and look of the date display, the calculation and number of religious holidays.

For a long time commercial software providers used national languages in menu navigation and user dialogs as a distinctive feature of different markets. Therefore, there is a trivial branching in the view paths when different localized variants of a program or operating system are available and can be offered.

As mentioned, there can be more than one view path available for a particular object type. On one hand this increases the probability of long-term successful access, with potentially higher costs however. Based on the object type and the possible required applications, further view paths result which also need to be realized. A simple example is the so called office packages and collections of different applications. You can work with a variety of formats – not only with the included subcomponents, but also through import filters with an array of further file formats.

These observations can help to identify an additional reference platform which is used only for a particular object but also be reproduced in other ways. For example, for the reproduction of a PDF, a particularly uncommon platform is not required if an equivalent viewer is available on a more commonly used platform. It become more difficult for aggregation if nothing is known about the alternative view paths and whether they reconstruct the object in question one hundred percent. A typical example is the import of Microsoft Word documents into other word processing applications. At this point, thoughts about user feedback already mentioned in a previous section could be taken into consideration.

There are certain costs associated with every View-Path which can be estimated for every step. If they cross a certain threshold, economic considerations could follow to the uptake of primary objects in the archive, such as declining particular data formats, since later reconstruction is expected to lead to high costs. Thus, depending on the situation

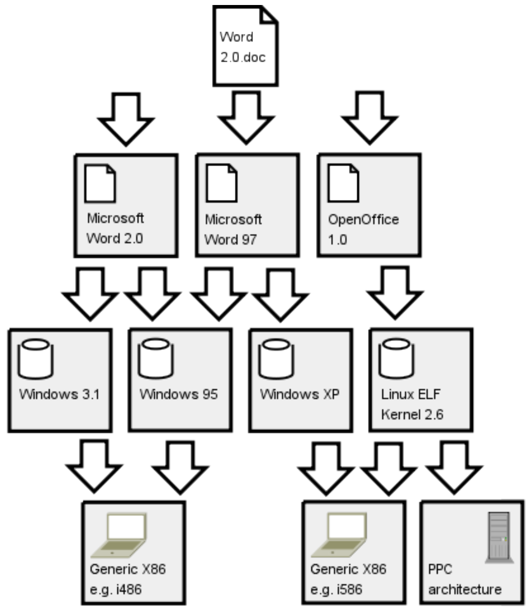

Figure 13: Many digital objects could be rendered in different ways and digital environments. Often more than one rendering application for several operating systems exists.

or archive, the maintenance of a certain View-Path could be challenged if good alternatives exist (Fig. 13). Consider a primary object of the PDF 1.0 standard which was created using a tool in a Mac-OS 7 desktop environment; it is not imperative that this environment is preserved. Under the following conditions the abandoning of a View-Path may be considered:

- There exist a sufficient number of further, equivalent and therefore simpler view paths.
- Due to the good and complete description of the format, it is easier to migrate a viewer for the respective current host system then to restore old utilization environments through emulation.
- This object type is the only one that requires a Mac-OS 7 environment.
- There is no special interest to preserve the utilization environment since it is not the focus of the library.

This approach can be extended to other view paths for an array of file formats and applications that require Apple or Microsoft operating systems. This is true only if, for example, no special requirement for a dedicated Apple utilization environment exists. This may be due to the type of user interaction, the look and feel of the graphical interface and the separate interest of applications. This can be demonstrated especially well with OpenOffice, which has been translated for several commercial and free Unixs like BSD, Solaris or Linux, MacOS X and for the Windows operating system. Here not all conceivable branches need to be kept in the archive. Similar is the case of the trivial multiplication of the view paths through various localizations. The case could be different if elsewhere secondary archive objects become the center of interest. Computer art or games are their very own evidence of their age.

Simpler and thereby often cheaper to maintain view paths can take precedence over others. However, it is difficult to predict the danger that a change of the reference environment has on the cost structure. On the other hand, precautions can be taken so that rare view paths at specialised institutes with specific collection tasks and corresponding financing continue to be looked after.

With the various strategies of long-term archiving, the emulators generate different costs on the side of archive management. *T*he stacked emulation leads to longer view paths with reduced need for migration. The costs arise from the increasingly complex access to the primary object. *T*he migration of emulators, UVMs and modular approaches generate shorter view paths with easier access to the primary object. However the cost lies in the regular maintenance of all required emulators or their subcomponents.

The first mentioned strategy is better suited for objects accessed rarely or institutions with small specific qualified user circles like archives or national libraries. The second strategy is preferred for object types that have often use and are for larger user communities.

The merging of certain view paths into a common environment could be useful for similar object types like images or for a simplification of View-Path creation for the user operation. For the investigation of computer games of a particular age, a producer or a certain genre, many applications can be placed together in a single container file of a particular emulator. This eases the exchange, since often the container alone distributes the configuration files and the start-up instructions of the emulator. This prevents the end user from needing to deal with the unfamiliar installation process. They can concentrate on a reasonable work distribution of their request. Some alternative aggregations are

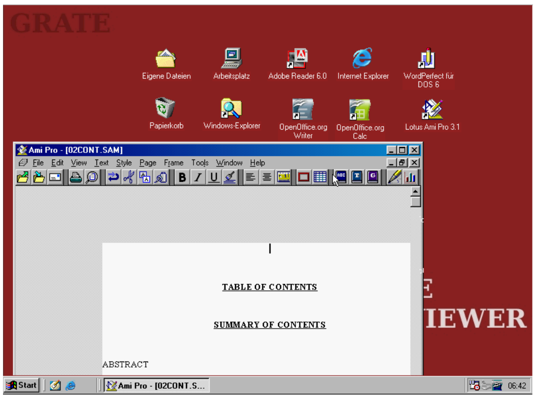

Figure 14: Migration-through-Emulation: Use a virtual printer to produce PDFs from deprecated text processing formats like AMI Pro. This is a strategy to omit non-existant export-filters within the original software or missing import filters in the actual software used.

also conceivable in order to access datasets of the public administration at later time points. These are typically in a compressed archived of combined data of elevation, statistics or analysis and comprises an array of typical object formats in a certain epoch. These could be stored in a shared container and contain helper utilities like the document printing by Postscript or PDF. Hereby virtual printer drivers can replace missing export functions in the old applications.

view paths in Technological Change

In the OAIS Reference model, the management of the digital long-term archive takes over a number of tasks which have to do with the lifecycle management of the primary object. For this the work process for the object reproduction is especially observed. For long-term management the task of preservation planning is particularly interesting.

A central moment is the regular control of the view paths with every change to the reference environment. Every platform change poses new challenges for the restoration of the environment. This inspection deals with an iterative process that runs through all registered object types in the archive. For this, a suitable strategy for the change over from one reference environment to another should be found. In addition it should be noted that additional file formats require new view paths.

Over time, the number of historical computer architectures inevitably increases so that with every change of the current working environment, higher migration costs for the applied emulators could follow. These boundary conditions influence the work that must be done on the side of the library or organization.

With the increasing perception of the digital long-term archiving as a new challenge, many new approaches could be proactively followed. It could, for example, be imagined that the manufacturers of operating systems already in the development could take certain hardware drivers for meta-devices into consideration. This could be virtual devices for graphic or audio output or the network interfaces. At this point even legal obligations could be in place to have such interfaces or drivers, which could work similar to the mandatory delivery of publications in many National Library laws of the world.

The mentioned challenges have clear economic implications for the archiving organizations dealing with the digital objects. They will have to cover part of the development of emulators and viewers or to buy this on the market. Many promising initiatives already exist that work with cooperative approaches. For unions like Nedlib or InterPARES it shouldn't be a (financial) problem to accept such challenges. If the target development of open standards and tools like PRONOM or Dioscurii result, the impact on individual organization will be limited and binding procedures could be developed that are supported by a large community.

Operation of Digital Archives

For a long-term digital archive this approach generated a set of tasks, starting with the identification of the object type when ingested into the archive, the generation, completion and storage of the required metadata and the regular checks on the existence or creatability of a rendering environment.

The classification of the ingested object and the generated metadata influence the needed steps for the later access. Beside this, the criteria of View-Path determination may vary depending on the targeted user group or archival institution. In general three major phases could be distinguished: The required work steps on the object ingest, the periodical operational procedures of archive operation and the guidelines for the object digest to the interested object user.

For the detection of digital object types, there are several strategies in place. The most prominent one at the moment is the PRONOM database and DROID tool (TNA 2007), (TNA 2008) of the British National Archives. This information might have to be extended with tool information like on applications or operating systesm along the View-Path.

Depending on the requested View-Path, further digital objects are needed. Because of certain object type's relatively high complexity, some considerations about how a View-Path is realised must be made. Especially for frequently requested paths, it could be conceivable to work with prepared environments. Considered together various requirements of the Archive operation result:

Creation of a background archive: In this case every single object needed to create a certain View-Path is permanently stored within the archive. These additional artefacts are to be kept like the primary objects of interest. At this point it could be considered whether the view path objects like the emulators, operating systems, specific helper software and description are bundled together in a single package or stored individually.

Operation of an online archive for direct access: For frequently requested secondary objects, it could be a good idea to store these in a special archive, additional to the long-term archive. This can relieve the load on the long-term archive and lead to a faster process of generating view paths.

Setup of a View-Path cache: For the often requested and more complex to reconstruct view paths, the use of a prepared working environment can reduce the work for users and archive operators. This cache could be a part of the online archive or be directly placed on the reference platform.

In the moment of digest from the archive, the procedure for the object rendering must be computed and executed automatically, or with the interaction of the archive user or archivist. These work steps are not simply copy and reproduce the objects bit stream, but actually allow an access to the object in a sensible way. At this point emulation and migration strategies do not differ much. The procedures for the object reproduction could be described by the aforementioned view paths. Additionally, independent view paths for the migrated object and the emulated may help to ensure authenticity. In every case the archive user needs a certain set of tools to access the object. These are often additional software which has to be kept in the archive too or in a specialized software repository.

Secondary Objects Software Archive

In general it could be considered impossible to open e.g. a PDF document only with computer hardware. For this they need at least a program for viewing, which on its part is not executed directly

Figure 15: Selection of ancient installation media of popular software of the decade before last for the X86 platform.

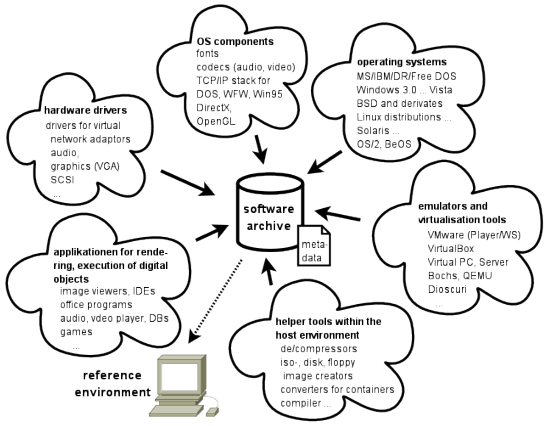

Figure 16: The X86 environment requires a number of software components besides the hardware emulators to be able to render/execute digital objects of different types, like the several applications, additional components like fonts and codecs, different operating systems, hardware drivers.

on the hardware and its interface for the file and display access, plus the retrieval of user details, is directly programmed (Fig. 15). This software requires an operating system that takes over the control of the input and output interfaces and realised the base user interaction.

Depending on the chosen emulation layer, additional software components may be needed (Fig. 16). While the emulation of applications for alternative access on a particular data format requirement for additional software is dropped, this increases with every lower layer in the software-hardware stack. Optimally the emulated applications run on a current platform, allowing direct access to the digital object or the respective format. As long as it manages to migrate the corresponding applications when a platform change occurs or create new ones if needed, this method for long-term archiving of certain file types is by all means attractive. It could be imagined that this method would be used for static data types like the various open and well-documented image formats. The emulation of an operating system should theoretically allow for all applications to be run on this operating system, with possible restrictions. Additional applications to be run are stored in a software archive next to the emulator for the operating system. When porting the operating systems' emulators, it must be ensured that the various applications in the software archive continue to function.

The hardware emulation sits at a very low layer (Fig. 7). Although this aims on a very general approach, it requires an entire array of additional components (von Suchodoletz, Rechert, and van den Dobbelsteen 2010). To actually be able to observe a given static object or to see a dynamic object run, the layers between the emulated hardware and the object itself must be bridged depending on the architecture. The bridge is realised by the above mentioned view paths. These differ depending on the chosen strategy, "Migration-on-Dissemination" or reconstruction of a utilization environment.

When reproducing a complete hardware through emulation, you require in every case at least one or potentially many operating systems which can be run as a base for the applications sitting on top. For a software archive this means that the operating systems must be kept along with the emulator for a platform. This is also true for the applications based on the operating systems needed for the display of various data formats. This is followed by a migration of the hardware emulator and the desired functionability of the operating system must be ensured. Since most applications work with the same principles as the operating systems, correct execution should follow directly from that operating system.

Best suited in terms of true long-term archiving are open source implementations of emulators. They allow the translation of the respective platforms as well as the long-term adaption of completely new architectures. Furthermore, it can be ensured that outdated peripherals are permanently reproduced and do not become a victim of the product cycle. A further point is the required adaption of the container file under circumstances where the guest system is installed. If the emulator changes the data format then these files are endangered exactly like other objects in their readability and interpretability (Fig. 17). However, commercial vendors usually provide import functions for previous versions. With free, open source emulators it can be ensured that a particular file format is frozen as an alterna-

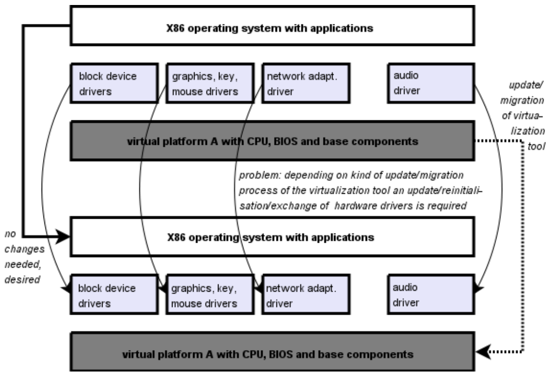

Figure 17: Depending on the kind of update, reinitialization of the guest operating system may be needed. This might require a completely new set of hardware drivers, which should be available in a software archive for emulation.

tive to further development.

Reference Environments and Remote Access A major challenge for the future access is the design of View-Path techniques for the rendering process. Appropriate reference environments try to present a wider spectrum of rendering environments in a rather compact and comfortable way. The base platform for them should be matched to the most popular operating systems and computer hardware of a particular timespan. This prevents the establishment and costly operating of a hardware museum and helps the user to orient oneself more easily in a familiar surrounding. Additionally, the reference environment should offer easy access to all metadata and required toolsets (von Suchodoletz and van der Hoeven 2008). But ev-

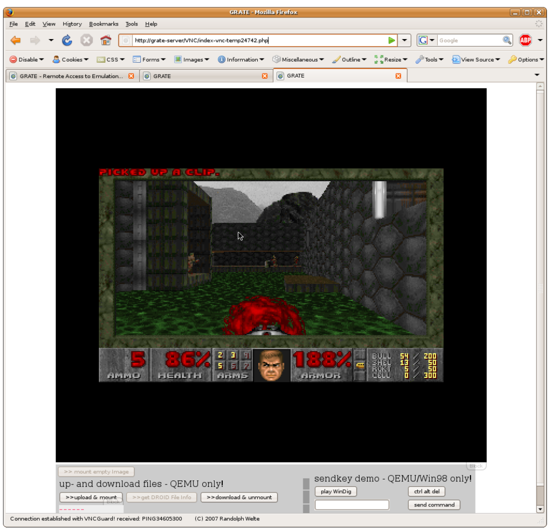

Figure 18: Transportation of the remotely rendered environment is done by GRATE, which offers a prototype of an centralized emulation service containing a number of different emulators and predefined view paths.

ery computer platform – the historical now have their own complexities and operational concepts – most future computer users will not find old user interfaces as easy to use as we might think today. The same is true for setup and installation routines of emulators and ancient operating systems. Another challenge arises from the transport of the requested artefact from the current into its original environment. Thus it would be desirable to centralize the whole process in specialized units with trained personnell and offer the services within a framework over the internet. This eases the complex procedures and allows them to be run on an average computer, reducing the functionality to a viewer in web-browser for example. The user gets the results presented via a virtual screen remotely on his or her computer.

This would help to prevent some shortcomings and circumvent a list of problems:

- The access to digital objects is not dependent on locally installed reference workstations e.g. in archives and libraries.
- The user does not need to be a specialist trained in ancient computer platforms.
- The management of such a service could be centralized and several memory institutions could share the workload or specialize on certain environments, sharing their expertise with the others.
- Problems of license handling and digital rights management could be avoided, because no item has to be copied onto users private machinery outside the premises of the services operator.
- Institutions like computer museums are able to present their collections in additional ways other than simply within their own room, thereby attracting more attention.

Within the PLANETS project (Farquhar and Hockx-Yu 2007), a pilot is being carried out by developing a prototype of an emulation service. This service is based on the available open source emulators mentioned above and allows them to run remotely. Transportation of the remotely rendered environment is done by GRATE (Fig. 18) which abbreviates *G*lobal Remote Access to Emulation Services (Welte 2009). With GRATE any user can easily access emulated environments via their internet web browser. The ideas of GRATE were extended into PLANETS networked services for "create view" and "migration" tasks (Rechert, von Suchodoletz, and Welte 2010).

Conclusion

The application of the long-term archiving strategy emulation achieves the preservation of almost any digital object over a long period of time. The most extensive and promising at the present time is the emulation of hardware, which allows for the resurrection of old utilization environments in which digital primary objects can be viewed or run in their unaltered form. Here, in only a few application circumstances is the the historic environment preserved as a regular working platform in which future users as well as the operators of the given period can use it. The main focus lies clearly on the "visualization" of primary objects and all the required tools and information. So that this can work long-term, an array of success terms must be considered – emulation counts to the more complex strategies which cannot be obtained for free.

Therefore, in order for successful access to a particular digital object at least one view path must be constructed. This normally begins with the creation of an alternatively functional and comparable application or the execution of the environment of the object, ending on the current platform with which the archive user interacts. This, as opposed to migration, is advantageous because the digital object itself does not need to be changed and thereby its integrity and authenticity is preserved. Two further direct application areas follow: (1) objects migrated several times into the current reference environment can, with a comparison to the original running in an emulated utilization area, potentially be discarded, and (2) "software archaeologists" obtain a digital toolbox with which they can analyse objects outside of the modern long-term archive. For the preservation of the view path, a software archive is required that must contain an array of important metadata like handbooks, instruction manuals and licence keys along with the software (von Suchodoletz, Rechert, and van den Dobbelsteen 2010).

Emulation cannot replace migration, but is instead a suitably complement to it. While migration processes of several static objects can largely be automated, emulation typically requires a high level of user interaction. Furthermore, in the emulator, migration is a conceivable strategy for long-term usability of these applications.

The extended understanding of dynamic digital objects does not mean that digital long-term archive needs to be reinvented, but that primary objects of interest could require certain secondary objects. For this, process determination for suitable format registries and other metadata schemas for the recording of additional required information for the handling of the respective object plays a role.

By now many metadata standards exist which offer, in addition to traditional archives, the ability to classify dynamic digital objects. However, standards may need to be extended with elaborate descriptions or installation manuals. Trustworthy digital long-term archives create new challenges for archiving organizations. The operations and ability of digital archives inevitably stem from traditional task descriptions: "New Skills Call for New Jobs". The similar is true for the archive operation itself; digital long-term archives are optimally based on open methods that make use of mutually defined and created standards. Experience shows that local and national solutions have not been sufficiently developed to meet the size of the task. Based on historic digital objects a high price in the form of high complexity with the access or the loss of information must be paid by proprietary developers. Big problems both for the primary and secondary object can at the moment be caused by the often discussed Digital Rights Management (Fig. 19). This serves in the respective "active" phase of an object the protection of unauthorised access or the upkeep of the rights of the original object creator. This rights management or copyright protection is in opposition to the archiving concern of digital long-term archives instructed to copy objects. Furthermore, the access to objects for later users should not be bound to the systems are probably not currently available anymore. At this point the legislator is requested to implement suitable conditions.

Figure 19: Without additional information stored with the metadata of the object, you might stick with access interface. The Lucas Arts game presented here needs authentication taken from a turnable three papers disk delivered with the software.

Background and Additional Sources

Parts of this paper present a condensed summary of (von Suchodoletz 2009) a dissertation finished in 2008. Some of the groundworks discussed here could be found in (van Diessen and van Rijnsoever 2002), (van Diessen 2002), (KB 2007), (Rothenberg 2000), (Wheatley 2001), (Reichherzer and Brown 2006) and in (Verdegem and van der Hoeven 2006). Information on the EU-funded PLANETS project is outlined in (Farquhar and Hockx-Yu 2007).

References

Becker, C.; Rauber, A.; Heydegger, V.; Schnasse, J.; and Thaller, M. 2008. A generic xml language for characterising objects to support digital preservation. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, 402–406. New York, NY, USA: ACM.

Bellard, F. 2008. Qemu - open source processor emulator. WWW document, *http://fabrice.bellard.free.fr/qemu*. [online; last accessed 11.01.2008].

DIOSCURI. 2010. Dioscuri java x86 emulator. WWW document, *http://dioscuri.sourceforge.net*. [online; last accessed 10.08.2010].

Farquhar, A., and Hockx-Yu, H. 2007. Planets: Integrated services for digital preservation. *International Journal of Digital Curation* 2(2).

Gabriel, D., and Ribeiro, C. 2001. A metadata model for multimedia databases. WWW document, *http://hdl.handle.net/10216/466*. [online; last accessed 10.08.2010].

KB, K. 2007. The archiving system for electronic publications: The e-depot. WWW document, *http://www.kb.nl/dnp/e-depot/dm/dm-en.html*. [online; last accessed 10.02.2008].

LoC, L. 2008. Premis - preservation data maintenance activity. WWW document, *http://www.loc.gov/standards/premis*. [online; last accessed 15.08.2010].

Lorie, R. 2002. *The UVC: a Method for Preserving Digital Documents - Proof of Concept*. PO Box 90407, 2509 LK The Hague, The Netherlands: IBM Netherlands, Amsterdam.

Oltmans, E.; van Diessen, R.; and van Wijngaarden, H. 2004. Preservation functionality in a digital archive. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, 279–286. New York, NY, USA: ACM Press.

Rechert, K.; von Suchodoletz, D.; and Welte, R. 2010. Emulation based services in digital preservation. In *JCDL '10: Proceedings of the 10th annual joint conference on Digital libraries*, 365–368. New York, NY, USA: ACM.

Reichherzer, T., and Brown, G. 2006. Quantifying software requirements for supporting archived office documents using emulation. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, 86–94. New York, NY, USA: ACM Press.

Rothenberg, J. 2000. An experiment in using emulation to preserve digital publications. WWW document, *http://nedlib.kb.nl/results/emulationpreservation-report.pdf*. [online; last accessed 05.06.2009].

Rothenberg, J. 2002. Avoiding technological quicksand: Finding a viable technical foundation for digital preservation. In *The State of Digital Preservation: An International Perspective, Conference Proceedings, Documentation Abstracts, Inc., Institutes for Information Science, Washington, D.C., April 24-25*, 1–32. 1755 Massachusetts Avenue, NW, Suite 500: Council on Library and Information Resources.

TNA, T. 2007. The technical registry pronom. WWW document, *http://www.nationalarchives.gov.uk/pronom*. [online; last accessed 10.01.2008].

TNA, T. 2008. Droid - digital record object identification. WWW document, *http://droid.sourceforge.net*. [online; last accessed 10.06.2009].

van der Hoeven, J.; van Diessen, R.; and van der Meer, K. 2005. Development of a universal virtual computer (uvc) for long-term preservation of digital objects. *Journal of Information Science* 31(3):196–208.

van der Hoeven, J. 2007. Dioscuri: emulator for digital preservation. *D-Lib Magazine* 13(11/12).

van Diessen, R., and Steenbakkers, J. F. 2002. *The Long-Term Preservation Study of the DNEP project - an overview of the results*. PO Box 90407, 2509 LK The Hague, The Netherlands: IBM Netherlands, Amsterdam.

van Diessen, R., and van der Werf-Davelaar, T. 2002. *Authenticity in a Digital Environment*. PO Box 90407, 2509 LK The Hague, The Netherlands: IBM Netherlands, Amsterdam.

van Diessen, R., and van Rijnsoever, B. J. 2002. *Managing Media Migration in a Deposit System*. PO Box 90407, 2509 LK The Hague, The Netherlands: IBM Netherlands, Amsterdam.

van Diessen, R. 2002. *Preservation Requirements in a Deposit System*. PO Box 90407, 2509 LK The Hague, The Netherlands: IBM Netherlands, Amsterdam.

Verdegem, R., and van der Hoeven, J. 2006. Emulation: To be or not to be. In *IS&T Conference on Archiving 2006, Ottawa, Canada, May 23-26*, 55–60.

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

von Suchodoletz, D.; Rechert, K.; and van den Dobbelsteen, M. 2010. Software archives as a vital base for digital preservation strategies. WWW document, *http://eprints.rclis.org/18764*. [online; last accessed 19.07.2010].

von Suchodoletz, D. 2009. *Funktionale Langzeitarchivierung digitaler Objekte – Erfolgsbedingungen des Einsatzes von Emulationsstrategien*. 37075 Göttingen: Cuvillier Verlag.

Welte, R. 2009. Funktionale langzeitarchivierung digitaler objekte – entwicklung eines demonstrators zur internet-nutzung emulierter ablaufumgebungen. Arbeitsgruppe am Lehrstuhl für Kommunikationssysteme, Universität Freiburg.

Wheatley, P. 2001. Migration - a camileon discussion paper. WWW document, *http://www.ariadne.ac.uk/issue29/camileon*. [online; last accessed 10.01.2008].