

Requirements for applying emulation as a preservation strategy

Jeffrey van der Hoeven; Koninklijke Bibliotheek; The Hague, The Netherlands

Bram Lohman; Tessella plc.; The Hague, The Netherlands

Remco Verdegem; Nationaal Archief of the Netherlands; The Hague, The Netherlands

Abstract

In 2007 the Koninklijke Bibliotheek (KB) – National Library of the Netherlands and the Nationaal Archief of the Netherlands delivered a durable x86 component-based computer emulator: Dioscuri, the first modular emulator targeted specifically at digital preservation. Recognised as a challenging but viable approach to retain long-term access to digital objects, the next step is to integrate this emulator with the operational digital archiving environment. This will offer a new and innovative long-term access solution based on emulation. To achieve this, both institutions recognised that several requirements have to be met: the organisation's policy and finance should support it; the significance of the digital collection and user requirements should be identified; software, documentation and user experiences should be preserved; preservation metadata should be captured; there should be a durable development process for emulation; and finally a technical infrastructure is needed to facilitate emulation as a long-term access strategy.

Introduction

With today's ongoing paradigm shift from physical documents towards digital, the term preservation acquires new meaning. Each day new inventions and technologies introduce better digital capabilities, quickly superseding older versions. Meanwhile, every individual and organisation builds up a digital history which grows in volume as well as in importance. Losing that digital information can have large cultural, social and financial effects.

As successful caretakers of physical artefacts, cultural heritage organisations were the first to be aware of this fragility. They started to research and develop digital preservation systems to secure storage of digital objects. However, preservation does not stop at careful storage but requires pro-active preservation policies and strategies to guarantee that digital objects will remain accessible. As the policy strongly depends on the organisation's nature, preservation strategies can be either focused on the digital object itself (migration), on the environment of that object (emulation), or a combination of these. Although the capabilities of emulation were disputed for a long time, awareness is growing that for some types of digital objects, such as interactive media, websites and complex database systems, emulation is the only viable approach [1].

In 2005, the Koninklijke Bibliotheek (KB) and Nationaal Archief of the Netherlands started a joint project to build, test and evaluate an emulator. Released in June 2007, the emulator, Dioscuri [2], implemented two key aspects: modularity and portability. Modularity provides the ability to easily emulate different target machines, while portability makes it possible to run

the emulator on different host machines, now and in the future. The KB and Nationaal Archief concluded that, although challenging, emulation is a viable approach to retain long-term access to digital objects [3]. This article proposes the necessary next steps for applying emulation as a strategy for long-term access to digital objects.

From tool to strategy

During the Emulation Expert Meeting held in October 2006, a group of experts in the field of digital preservation, IT and emulation identified the next steps necessary to successfully apply emulation as a digital preservation strategy [4]. Based on these steps and the experiences gained during the development of Dioscuri, the KB and Nationaal Archief defined the following six requirements that should lead to successful application of emulation as a long-term access strategy:

1. take into consideration the organisation's policy regarding preservation of digital objects and find structural financial support for long-term access strategies;
2. identify the significant properties of the digital collections and gain insight into user requirements;
3. preserve old software, documentation and user experiences;
4. define the necessary preservation metadata;
5. set up a durable emulator development process;
6. create a technical infrastructure for emulation.

Policy and financial support

The motivation to preserve a digital collection should be a reflection of the organisation's policy. For memory institutions like libraries, archives and museums digital objects of any kind should in most cases be preserved for the long term, from decades up to centuries. But preserving digital material, even for a couple of years, is challenging due to rapid IT developments. In any case, a clear preservation policy with subsequent strategies to retain access to digital content is required. The KB has a long-term commitment to preserve all published material, including electronic publications, for future generations [5]. The Nationaal Archief has a legal obligation to preserve governmental records [6]. Both organisations recognise the need for preservation strategies and perform research into a wide variety of preservation topics such as emulation, migration, file format characterization, metadata and web archiving.

A common misconception is that applying migration as preservation strategy is less expensive than emulation on a per-object basis. Usually not factored into the equation is that an emulator such as Dioscuri tackles multiple formats at once, and does so for the foreseeable future. Migration will continually need to be carried out to update objects to current versions. A comparison of long-term costs is depicted in figure 1 [7]. This figure shows the actual project costs of developing Dioscuri

compared with estimated migration costs required for retaining access to ten million digital objects (each object about 1 megabyte in size) over a period of twenty-five years. The calculated costs for preserving a single digital object of that size in the KB's e-Depot is about € 0.015 per year.

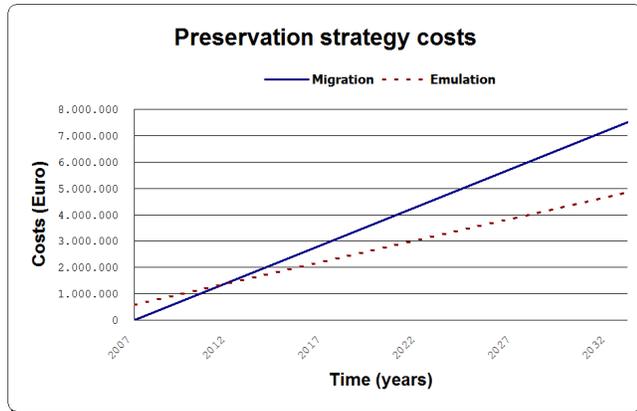


Figure 1: estimated costs for preservation strategies. The costs of preserving one digital object of 1 megabyte in size in the e-Depot of the KB are € 0.075 per five years. This includes expenses for storage and server capacity but excludes personnel costs, training, housing, etc.

Applying migration to ten million objects once every twenty-five years, costs five million Euros. The actual expenses for emulation developing Dioscuri cost about 600K euro. Taking into account maintenance costs of 20K euro a year, both preservation strategies become break even after five years; after that emulation becomes the more cost-effective solution.

Significant properties and user requirements

To understand the value of a digital object, its significant properties have to be identified. For example, a significant property of a Microsoft Word document is the formatted text it contains. For a music record it is the sound. For a website it is more difficult as text and layout are important as well as the ability to navigate through the site. In the latter case, the object is not 'static' anymore, but has a dynamic component that is also important to preserve. A method for identifying these properties is by looking at five attributes: context, content, structure, appearance, and behaviour [8]. Each digital object contains these attributes, although the importance of them will vary. Referring to the Microsoft Word document, its content, structure and context may be more important than appearance and behaviour, while the website definitely requires a proper layout and functionality. These attributes directly relate to the preservation strategy that is to be applied. Migration is useful for preserving the first three attributes, but less qualified for maintaining the original appearance and behaviour of the object. Instead, emulation retains both the original object as well as its native environment and thereby offers minimal risk of affecting the authenticity.

Preservation of software, documentation and user experiences

At the start of the emulation project, it was decided to build an emulator in software to emulate hardware[1]. Although this "software-emulation-of-hardware" approach can be quite complicated, it is more straightforward than emulating higher levels, such as an operating system (OS) or a user application. Because hardware specifications are well defined, this functionality is easier to reproduce than that of an OS or application software. Moreover, this approach retains the original functionality of the OS, applications, drivers and configuration, which secures authenticity of the original computer environment. The drawback is that this software has to be preserved as well.

Keeping in mind that preservation strategies will be applied over the long term, knowledge of today's common computer environments is likely to be as unfamiliar as is working with WordPerfect 5.1 nowadays. To cope with this problem, documentation such as hardware specifications, manuals, tutorials, source code and experiences of users operating this software should be kept safe. During the Dioscuri project, a first step was taken by collecting important hardware specifications and recording differences in specification and practice. Furthermore, the European project Planets is developing use cases to describe how a user is working with a selected number of computer environments [9]. Eventually, this should be enough to recreate a picture of today's state-of-the-art computer experience.

Preservation metadata

To make sure that accurate emulation renditions can take place in the future, metadata for each digital object are required. An absolute minimum of metadata elements are: file format, timestamp and operating system. In most cases, the file format denotes the application used to view the object. The timestamp gives an indication of the common software and hardware used during that time, while the operating system defines most of the underlying platform and supported applications on top of it.

Still more contextual information is required to know what the actual computer environment should look like. However, it is not required to preserve this additional information on a per-object basis. Instead, a metadata registry could serve as a database describing the dependencies between object-software-hardware, often called viewpaths [10]. To render a Lotus123 spreadsheet document in its authentic environment one should know on which operating system the Lotus spreadsheet program used to work and which hardware it required to run the OS. Therefore, metadata should be preserved describing which digital file format requires what software, which in turn requires what hardware. Several metadata registries are available nowadays to keep track of these dependencies. Since 2004, the KB has the preservation manager[11], a stand-alone application dedicated to this task. Other software is under development, like PRONOM [12] and GDFR [13]. The recorded minimal set of metadata for each object serves as a key to the correct viewpath in the registry.

Durable emulator development process

A software product has an average lifetime of about ten years [14]. During this rather short period, software will be patched, upgraded and abandoned altogether. Emulation only becomes

useful decades after the original environment has become obsolete - any earlier and current hardware can usually be used to run the original objects. For emulation to be successful, a durable software development process must be applied. There are several factors to consider:

- the underlying hardware will continue to change;
- the operating system used to run the emulation software will also continue to evolve; and,
- the emulation software itself.

Hardware

Hardware continues to evolve at an ever faster rate. Processing units, memory and video cards continue to aim for better performance; storage methods, be it hard disks or compact-disc based methods, as well as others, continue to increase in storage size. Although no major paradigm shifts have taken place recently, these technologies will at some time be pushed to their theoretical limits, and new methods will need to be developed to continue the ever increasing evolution. It is these shifts that pose the greatest threat to digital preservation. Once a platform has become completely obsolete, migration is no longer viable, and emulation is the only option.

Operating system

The operating system on which the emulator runs will be directly influenced by the underlying hardware. But for the emulator this will be a smaller change than that in hardware. Several strategies exist to address this issue, such as emulator 'migration'; emulator stacking; or using a virtual layer [1]. The latter solution has been applied by the Dioscuri project, where Dioscuri uses a Java Virtual Machine (JVM) as intermediate layer between platform dependent operating system and platform independent emulator. This approach has as additional advantage that the emulator becomes portable, so in case one platform becomes obsolete, the emulator can still be run on other operating systems that support a JVM.

Although this layer of abstraction requires extra computational time, the benefits become apparent in the support over the years. Changes to the operating system will only need to be taken care of in the virtual layer. Although this might seem like a trade-off, it means that the emulator itself can virtually remain unchanged for years, as the virtual layer supports it.

Emulation software

As hardware technology continues to evolve, the need to evolve the emulator itself also becomes apparent. New devices and technologies will need to be incorporated in the emulator, preferably before the physical hardware is obsolete. Only then the emulator can be tested and validated against the original hardware it aims to substitute. The emulator should provide easily extendable support for new technologies and components. Software development should proceed along object-oriented lines, as this will provide the most flexible way of replacing and upgrading components, much like current hardware. The Dioscuri emulator is developed in a modular structure, mirroring the hardware it emulates, and is as configurable and upgradeable as real hardware. New developments in protocols, standards and internal and external components can be easily assimilated in the application.

Interfacing with hardware requires low-level knowledge of the components and behaviour. For faithful reproduction of the original platform, hardware specification documents must be available. Unfortunately, not all of these documents are publicly available, probably because they are owned by commercial companies, which are not eager to release this information for fear of benefitting the competition. In that case only via questionable means, such as reverse engineering, this knowledge can be obtained; and even then it is often incomplete or incorrect. As a result the emulation of these components can be fraught with errors. However, international standards, such as the Advanced Technology Attachment (ATA) [15], do not have this problem as descriptions are publicly accessible.

Additionally, the emulation program code (and preferably source code) also need to be documented and stored. All of these need metadata associated with them which indicate the platform, hardware and software requirements to run them.

Technical infrastructure: the next generation emulation-based access

Considering the requirements discussed in the previous sections, the final step for applying emulation as a digital preservation strategy is to combine all technical components into an emulation infrastructure. This infrastructure should be capable of rendering the digital object at the time the object is requested from the repository. The user should not be bothered with any technical issues around the emulation process. Instead, rendering services should enable the user to experience the digital object in different manifestations. In figure 2, a schematic decomposition of this infrastructure is depicted.

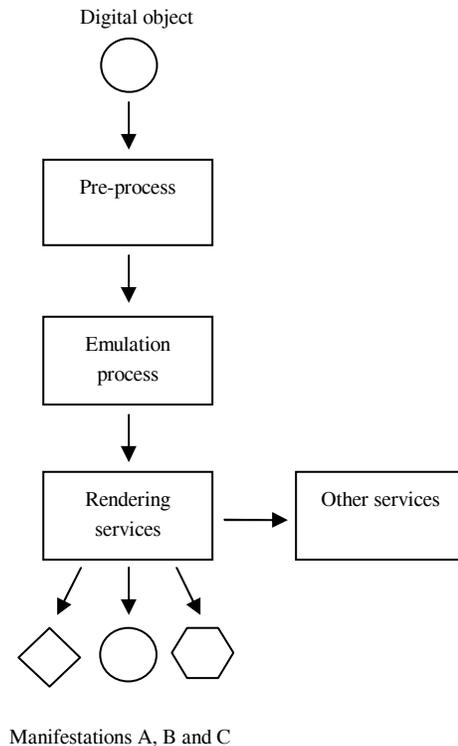


Figure 2: schematic decomposition of infrastructure

Pre-process

The infrastructure starts with a digital object that is being prepared for emulation. This step is called the pre-process and entails the compilation of the targeted hardware and software environment. It is unwieldy to have to recreate the original environment manually for every request. Therefore, an automated pre-process, transparent to the user, needs to be put in place to make the approach scalable and convenient to use. The computer environment will consist of the emulator, the operating system, the application(s), additional dependencies like font libraries, codecs and plug-ins and the digital object itself [16]. Using the preservation metadata of the digital object as a key to the metadata registry, the original hardware and software dependencies can be traced. However, multiple viewpaths may arise as the object in question may have multiple rendering options at that time. To define which viewpath is the preferred one, a resolving system is needed to make a 'best-fit' match between the required and available environments. To achieve this goal, the resolver requires a method for comparing environments consisting of:

- a scoring model for giving a score to each component (application, operating system, hardware) of the required environment that provides a quantitative way of comparing the components; and
- weights for each component, to determine the importance of the component as part of the whole environment.

Note that these specifications indicate the minimally required environment, but more environmental aspects may be added.

The following example provides an insight into the resolver's functionality:

A user would like to view a Microsoft Word document version 6.0, which requires MS Windows 95 and an Intel 486 platform.

Emulation environment A offers MS WordPad as text processing application, MS Windows 95 as operating system and an emulated Intel 486 platform. MS WordPad is not very accurate at opening MS Word documents and gets a score of 40% out of 100%. The OS and hardware are exactly as required so they receive a full 100% score each.

Emulation environment B offers MS Word 7.0, MS Windows 98 and an Intel Pentium platform. MS Word 7.0 can open Word documents reliably and therefore receives a 90% score on application level. The OS is very similar and receives 80%, while the hardware is also a very close match, 80%.

Given weights of 0.5 for application, 0.3 for OS and 0.2 for hardware, the following results can be found:

Emulation environment A:

$$40\% \times 0.5 + 100\% \times 0.3 + 100\% \times 0.2 = 70\% \text{ match.}$$

Emulation environment B:

$$90\% \times 0.5 + 80\% \times 0.3 + 80\% \times 0.2 = 83\% \text{ match.}$$

The preference in this example goes to Emulation environment B. Note that the scores are indicative and in this example heavily weighted towards the application level.

Once all hardware and software components are uniquely identified, a software image has to be compiled. A software image is a digital representation of a floppy, hard disk, CD-ROM or other medium containing a file system with files. For emulating the original environment correctly, all software components should be included in this image. Several methods for compiling this image can be used:

- Store a disk image containing pre-installed application software and operating system. This is the fastest way for emulators to create a target environment, but requires a very large amount of storage space as each object requires a unique disk image.
- Store a simple set of images containing only an OS on which the application software can be installed on demand. Before the emulator can recreate the target environment the required software needs to be installed. Although this usually requires user interaction, it is possible to automate this. However, it greatly reduces the storage space necessary.
- Store a simple set of images containing only an OS, along with a differential bit stream containing information of application software installed onto that image (this can easily be created by 'subtracting' an OS-only image from the OS plus application image). When required, the environment can be recreated by merging the differential bit stream with the original OS image.

The advantage of the latter method is that it requires no user interaction to create the environment, and saves storage space (figure 3).

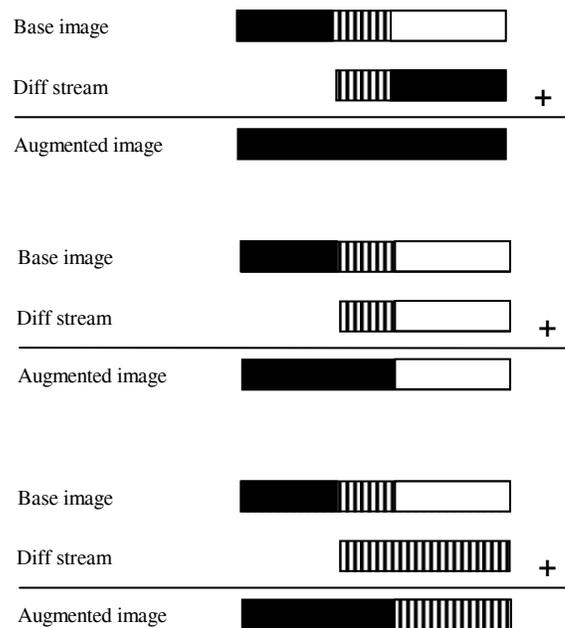


Figure 3: Augmented disk image approach

Emulation process

After preparation of the required software, the emulation process has to be configured and executed. Based on the preservation metadata and the scoring-weighting method, the selected emulator has to be set up. In case of Dioscuri, each module can be configured using an XML-based configuration file. From then on, an invocation of the emulator activates the emulation process. Depending on the architecture of the infrastructure, the process either runs at the client or via a client-server architecture where the virtual input and output devices such as screen, keyboard and mouse are running at the client while the actual processing is done on a remote server. For user convenience the latter approach seems favourable as emulation processes tend to be quite resource demanding. This remote emulation approach is currently researched by the European project Planets [9].

Rendering services

An often cited disadvantage of emulation is the lack of ability to conveniently reuse existing information contained in the original digital object. This information is “trapped” insight the emulation process and is difficult to extract and copy into a current computer environment. To support this kind of functionality, rendering services should be developed offering vernacular extractions of the original object [17]. These could be content extraction services such as copying images, text, sound, or data sharing in which current computer files can easily be swapped in and out the emulated environment. Moreover, user sessions could be recorded allowing a user to restore rendered environments from earlier sessions. A first attempt of vernacular extraction in Dioscuri is the ability to extract text from an emulated document via the common clipboard of the OS into text processing software of today.

Instead of rendering objects for a human, rendering services could also be used by other services. Extracted content could be automatically incorporated into a new document. This approach can be seen as a combination of emulation and format migration.

Conclusion

The emulation project of the Nationaal Archief and KB delivered the world’s first modular emulator for digital preservation: Dioscuri. Based on knowledge gained during this project, six requirements have been defined that are necessary to successfully apply emulation as a preservation strategy. These requirements are: an organization-wide preservation policy and financial support; the significance of the digital collection and user requirements should be identified; software, documentation and user experiences should be preserved; preservation metadata should be captured; there should be a durable development process for emulation; and finally a technical infrastructure is needed to facilitate emulation as a long-term access strategy.

In the near future, both organisations are willing to further implement these requirements starting with Dioscuri as emulator and integrating it into an emulation infrastructure. This infrastructure should be capable of rendering a wide range of digital objects currently preserved by KB’s e-Depot and the planned Digital Depot of the Nationaal Archief. Taking user requirements into account, this infrastructure is envisioned to offer

various rendering services to ensure that the wisdom of today will be the foundation of knowledge for future generations.

References

- [1] R. Verdegem, J.R. van der Hoeven, Emulation: To Be or Not To Be, *proc. IS&T Archiving 2006*, pg. 56. (2006).
- [2] Dioscuri, the modular emulator for digital preservation (2008). Available at: <http://dioscuri.sourceforge.net> (accessed April 2008).
- [3] J.R. van der Hoeven, B. Lohman, R. Verdegem, “Emulation for Digital Preservation in Practice: The Results” *J. The International Journal of Digital Curation*, 2, 2 (2007).
- [4] Statement Emulation Expert Meeting (2006). Available at: http://www.kb.nl/hrd/dd/dd_projecten/projecten_emulatie-emstatement-en.html (accessed April 2008).
- [5] e-Depot, Koninklijke Bibliotheek, The Hague, The Netherlands (2008). Available at: <http://www.kb.nl/dnp/e-depot/e-depot-en.html> (accessed April 2008).
- [6] Regeling Geordende en toegankelijke staat archiefbescheiden (*Regulation on the Arrangement and Accessibility of Records*, 2002). Available at: http://www.nationaalarchief.nl/images/3_2563.pdf (accessed April 2008).
- [7] E. Oltmans, N.J.C. Kol, “A Comparison Between Migration and Emulation in Terms of Costs”, *J. RLG Diginews*, 9, 2 (2005).
- [8] J. Rothenberg, T. Bikson, Carrying Authentic, Understandable and Usable Digital Records, Report To the Dutch National Archives And Ministry of the Interior, The Hague, The Netherlands (1999). Available at: http://www.digitaleduurzaamheid.nl/bibliotheek/docs/final-report_4.pdf (accessed April 2008).
- [9] European project Planets (2006). Available at: <http://www.planets-project.eu> (accessed April 2008).
- [10] R.J. van Diessen, Preservation requirements in a deposit system, IBM/KB Long-term preservation study report series number 3, The Hague, The Netherlands (2002). Available at: http://www.kb.nl/hrd/dd/dd_onderzoek/reports/3-preservation.pdf (accessed April 2008).
- [11] Preservation Manager (2004). Available at: http://www.kb.nl/hrd/dd/dd_onderzoek/preservation_subsystem-en.html (accessed April 2008).
- [12] PRONOM (2008). Available at: <http://www.nationalarchives.gov.uk/pronom/> (accessed April 2008).
- [13] GDFR (2008). Available at: <http://hul.harvard.edu/gdfr/> (accessed April 2008).
- [14] T. Tamai, Y. Torimitsu, Software Lifetime and its Evolution Process over Generations, *Proc. Software Maintenance*, pg. 63. (1992).
- [15] Advanced Technology Attachment (ATA). Available at: http://en.wikipedia.org/wiki/AT_Attachment (accessed April 2008).
- [16] G. Brown, T. Reichherzer, Quantifying Software Requirements for Supporting Archived Office Documents using Emulation, *Proc. JC DL 2006*. Available at: <http://www.cs.indiana.edu/~treichhe/jcdl2006.pdf> (accessed April 2008).
- [17] J. Rothenberg, Emulation: context and current status, The Hague, The Netherlands (2003). Available at: http://www.digitaleduurzaamheid.nl/bibliotheek/docs/white_paper_emulatie_EN.pdf (accessed April 2008).

Author Biography

Jeffrey van der Hoeven started his work in the field of digital preservation during his graduation assignment on the Universal Virtual Computer (UVC) at IBM Netherlands N.V. in 2003. In 2004 he obtained his Master degree in Computer Engineering at Delft University of Technology and started his career at the Digital Preservation Department of the Koninklijke Bibliotheek. He conducted research into emulation-based

preservation and joined the emulation project in 2005. Since then he has been involved in various other projects as well like the European projects Planets and PARSE.insight.

Bram Lohman graduated from Delft University of Technology in 2001 with a Masters degree in Electrical Engineering. In 2004 he joined Tessella plc. as a software engineer, and has been involved in digital preservation since 2006. He has been actively developing the Dioscuri emulator at the Nationaal Archief since joining the emulation project, and has been involved with the Nationaal Archief's Digital Depot early 2008.

Remco Verdegem began his professional career in the area of information technology in 1989. In October 1998, he joined the Dutch State Archives' Service, where he was among other things responsible for the functional maintenance of the archival system for paper records. From October 2000 till July 2003 he was the project manager of the Digital Preservation Testbed project. Since April 2005 Remco is working as Senior Advisor Digital Longevity at the Nationaal Archief, where he has been involved in a variety of projects, like Dioscuri, the European project Planets and the digital depot project of the Nationaal Archief.