Het Nieuwe Instituut

Design

Architectuur

Digitale Cultuur

EaaS as a preservation strategy for HNI — Research into the feasibility of Emulation-as-a-Service

author
Henk Vanstappen
(DATABLE)

date
14 May 2020

Contents

I. Management summary

This report explores the possibilities of emulation as a preservation strategy, focusing on the context of Het Nieuwe Instituut (HNI) as an architecture archive. The results must enable HNI to form an idea of the possibilities that emulation offers, where this can be applied and what requirements this imposes on the organization in terms of technical infrastructure, work processes and expertise.

In the first chapter, we define the concept of emulation and emulation as a service. The second chapter explores the application of emulation as a preservation strategy, where technical, as well as organizational and legal issues and opportunities are discussed. The chapter concludes with a general overview of tasks when an organization implements emulation as a preservation strategy. Chapter 5 then focuses on the implementation of emulation in HNI relating this to HNI's preservation policy and infrastructure, and illustrated by some implementation examples of three file formats.

For the HNI's born digital collection, emulation is an option worth considering. As a complementary or in some cases even an alternative strategy. The knowledge and technology to make digital objects (files or full disk copies) accessible in an emulation environment is available, but often still fragmented and developed in temporary projects or by organizations whose sustainability can be questioned. A service model in which public cloud emulation environments are offered to multiple organizations can realize economies of scale and thereby achieve a more sustainable ecosystem.

It is therefore an attractive option for HNI to be part of a – yet to be realized – partnership that offers this service model. In that case too, however, HNI still has some issues to tackle and choices to make. These choices will have to be part of the global preservation policy and elaborated in the file format strategy. On a technical level, this strategy will have to be translated into requirements with regard to the digital repository management system, procedures and work processes. Finally, HNI must formulate a policy with regard to the legal feasibility and possible risks of acquiring, using and offering software that is still copyrighted.

2. Introduction

Het Nieuwe Instituut (HNI) is increasingly acquiring digital archives of architects, urban planners and organizations working within the design domain. This requires new storage strategies and work processes with regard to the sustainable management of digital objects. To prevent loss and unwanted changes of information, a safe environment is required that safeguards the integrity of the digital objects. This environment also contributes to the preservation of the authentic information objects.

The accessibility of digital object is closely related to the technical environments they were created on. As physical and software components of these environments become obsolete, they are gradually replaced. On a long term, the resulting new environments are unable to render digital objects the same way as when created. Meanwhile, the old hardware inevitably becomes obsolete and becomes useless due to wear, making it impossible to use the outdated software components. Due to the complexity of the design software file formats, converting (migration) to another more sustainable file format is often a problematic strategy. That is why the HNI is investigating the possibilities of emulation. This is a preservation strategy aimed at preserving digital resources as originally created and used. By applying emulation techniques it becomes possible to mimic the behavior of an outdated computer on another (newer) computer.

An emulation strategy has to be translated into workflows, actions and a preservation planning. This is supported by an infrastructure with certain functions, metadata, protocols and standards, requiring specific knowledge and skills. This report explores the possibilities of emulation as a preservation strategy, within the context of HNI as an architecture archive. The findings can enable HNI to get a better understanding of the possibilities emulation offers in relation to its preservation policy, how emulation can be applied and what requirements this imposes on the organization in terms of technical infrastructure, work processes and expertise.

3. Emulation and Emulation as a Service (EaaS)

Emulation can be described as the imitation of a certain computer platform or program on another platform or program. In more technical terms, it can be defined as “the re-creation on current hardware of the technical environment required to view and use digital objects from earlier times.” (Holdsworth & Wheatley, 2001). As opposed to a migration strategy in a digital preservation policy, emulation might be defined as the logical replication of a computer system, not of the file itself (Rechert, 2016).

As explained in van der Hoeven (2005), different levels and forms of emulation exist, such as software emulation, system emulation and hardware emulation. In full emulation, the emulator runs on top of the host operating system, instead of running directly on hardware. Therefore we define an emulator here as a software program that runs on a host platform (hardware and operating system) and virtually recreates the hardware target platform (van der Hoeven, 2005). Emulation can be effectively applied in different scenarios, such as running an operating system meant for other hardware (e.g. console-based games on a computer), running software fit for another operating system or running legacy software after comparable hardware has become obsolete.

An emulator creates a virtual machine (VM) which resembles (and is often confused with) the process of virtualization. The difference with virtualization is that virtualization accesses hardware directly, while emulated environments needs a software bridge to interact with the hardware. Virtualizers (such as VirtualBox, VMWare or virtual machines offered by Amazon[1], Azure[2] and Google[3]), are only able to run on contemporary hardware (i.e. Intel-based x86 virtual machines[4]), whereas emulators may be used for almost any technical platform, including obsolete ones (Rechert, 2016).

A consequence of the use of the software bridge is that it requires a lot of processing power of the host environment (More, 2014), while virtualization utilizes the underlying hardware directly, thus offering (almost) native execution speed.

Virtualization can be useful as a means to support multiple environments, each configured for a specific software/operating system stack. All these so called runtime environments can be hosted by multiple virtual machines on a server as long as these conform to the same architecture as the underlying hardware. This said, virtualization is a feasible strategy, but will last only as long as the appropriate hardware doesn’t become obsolete.

A common approach to emulation is to create a disk image, which can be loaded in the emulator. This disk image contains a stack of software components, suitable for running software which eventually can open files in a supported file format. The components typically include a configured operating system, with

[1] Amazon Elastic Compute Cloud, https://aws.amazon.com/ec2/

[2] Microsoft Azure Virtual Machines, https://docs.microsoft.com/en-us/azure/virtual-machines/

[3] Google Compute Engine https://cloud.google.com/compute

[4] https://en.wikipedia.org/wiki/X86

all the necessary system software (e.g. software libraries) so it can run the appropriate application. The application can either run autonomously (e.g. a game) or will open the original files. These disk images can be prepared for each (set of) files in a particular file format, which are separately loaded onto the user's request. Alternatively, a disk image can be made that already contains both the software environment and the files themselves. In some cases these disk images may be created directly from the original working environment on existing hardware (e.g. the architect's personal computer).

The advantage of the use of a disk image with a complete environment is that no configuration is needed when a user wants to access the digital object. To achieve this, it must be ensured that the software environments (in disk images) can run 'anywhere, forever' (Rechert, 2016).

These disk images can be made available and accessed in different ways:

- loaded on a local machine in the emulator
- stored on a local machine and loaded (via a web browser) on an emulator that runs on a remote server (semi EaaS)
- stored on a remote server and accessed via a web browser (i.e. full EaaS solution).

Maintenance of the emulation environment can be provided as a service model, which allows organizations to access emulated environments and make them available to users, e.g. via a web browser. By analogy with SaaS (Software as a Service), this service is called Emulation as a Service or EaaS. For the archival institution it reduces much of the legal and technological complexities. As resources are shared, EaaS may also optimize costs. The main advantage of EaaS is that it simplifies access to digital objects and provides customized environments to users. The EaaSI project advocates a highly automated environment, in which a process is started when a file is selected to be opened for emulation. In a first step of this process, the emulation service receives the file, identifies it and extracts metadata such as the creation and modification dates. With this information, the service selects the most appropriate emulation environment with the appropriate application. The environment then opens the file for th user to interact with. This approach requires a highly degree of documentation of both software and file formats, for which EaaSI counts on data from PRONOM and hosted by Wikidata.

4. Emulation and EaaS as a preservation strategy

4.1. Research projects & initiatives

Nowadays, emulation is mainstream technology, but its use for supporting legacy software and legacy operating systems and obsolete hardware is still limited. In recent years however, emulation has gained more attention from the preservation community as a viable alternative for file format migration as a preservation strategy (Rosenthal, 2015). Migration is said to be an object-centric approach, focusing on the intellectual content of the document (Allen, 2018). It can be quite easily applied to relatively simple objects such as text documents and images. Emulation, on the other hand, can be considered as a way to preserve the digital object as an artifact, within the context in which it was created.

Various initiatives have been launched in recent decades to investigate the feasibility of emulation for preservation. Many of these research projects led to the development of test environments and tools.

The aim of CAMiLEON (Creative Archiving at Michigan and Leeds) was to assess emulation as a digital preservation strategy. The project evaluated publicly available emulators, explored emulator development, conducted test cases and undertook cost-benefit analysis of emulation vs. other digital preservation strategies (Granger, 2000).

During the PLANETS project a prototype GRATE was developed, which wraps QEMU and Dioscuri emulators into a single networked application. The prototype was followed by GRATE-R, the PLANETS remote emulator service (von Suchodoletz and Cochrane, 2011). GRATE-R has an emulation workflow engine, where a user clicks on a file he wants to open and GRATE-R returns the emulated file back on the user interface (von Suchodoletz, 2010a).

The KEEP project ran from 2009 to 2012 and produced an emulation framework. Over 30 file formats were supported on multiple platforms (x86, C64, Amiga, BBC Micro, Amstrad, Thomson TO7) (Schmidt e.a., 2010). The University of Freiburg developed bwFLA to provide an EaaS solution.[5] The communication between the emulator and the user takes place via standard HTTP. Thus there is no need for a user to install software, or browser plugins. The EMiL (Emulation of Multimedia-Objects in Libraries) project (2014–2016) extended the bwFLA project to simplify usage for multimedia objects in libraries and museums.[6] Olive, developed at Carnegie Mellon University contains a client software (VMNetX) that can execute virtual machines directly from any webserver.[7] The framework underlying the Internet Archive's software library emulators run not on some other computer accessed via the network, but on the user's own computer using JavaScript. So the IA framework does not require any special software on the user's machine. Nor, unlike bwFLA or Olive's "cloudlet" configuration, does it require the archive to pay for infrastructure to run the emulations, merely to store them and serve

[5] http://eaas.uni-freiburg.de/

[6] https://github.com/emil-emulation

[7] https://olivearchive.org/

them to user's browsers. The Internet Archive uses among others emulators from the MAME project, and presents a.o. a CAD software for an early Apple IIe.[8] The Internet Archive service allows for running games and other software applications, which gives a fairly good impression of the original software functionalities and look and feel, but input/output of files is not supported. Finally, the EaaSI (Emulation-as-a-Service Infrastructure) program is focused on the development of technology and services to further expand the capabilities of the EaaS software.[9] The EaaSI program aims to establish a community of partner institutions using the EaaSI software platform. It will also implement a resource sharing functionality, which enables distribution and retrieval of emulation environments and software installation media between users. To improve the description and discovery capabilities, the project will contribute to data in the Wikidata body of knowledge (via the WikiDP portal).[10] Finally it will prototype various modules and services for management of end-user access, including the Open Source Software Sandbox. This sandbox offers access to several emulated environments (including a FreeCAD application running in a Ubuntu 10.10 operating system). For copyright and end user license agreement issues, the sandbox is restricted to environments with open source software.

4.2. Use cases for emulation as a preservation strategy

Emulation may be applied at several phases in a digital content management workflow that involves obsolete software and file formats (Cochrane, 2014):

- Appraisal and selection: an emulation environment may be used at the very beginning of a workflow for processing a born digital archive, i.e. at the pre-ingestion phase. After this phase in which the files are explored in their original form, a migration strategy may still be applied to the selected files.
- Preservation: an emulation environment may be an indispensable step in a migration process, e.g. when a file in an obsolete format can only be opened in software that runs on obsolete operating systems and hardware platforms only.
- Access: in a full emulation strategy, users have access to the original file, inan environment that is close as possible as the original environment. Again, this may be in parallel with a migration strategy, where files are made available in both the original and a normalized format.

End user access in an emulation environment can be motivated by the need to view and assess a file in its original format and in its original environment. Reasons for this are, for example, a more genuine user experience (e.g. for researchers or in an exhibition setup) or the need to prove the authenticity of a file (De Vos, 2020).

In many cases however, emulation may even be the only adequate approach for accessing digital files, either because there is no suitable software available that runs on contemporary operating systems and hardware. This is especially true

[8] https://archive.org/details/090_CAD_Draw

[9] https://www.softwarepreservationnetwork.org/eaasi/

[10] https://wikidp.org/about

for complex files that depend highly on the software they were created with, and therefore cannot be migrated to another format without loss of significant properties (information, interaction or user experience). For preservation of applications such as games or artworks, emulation often is the only strategy that preserves behaviour and interaction in a sufficient way. It is not a coincidence that most of the emulation applications have been developed for legacy computer games.

4.3. Opportunities and issues

Preservation is about keeping access to the contents of a file for an extended time. Emulation may be a promising strategy, but it also has a number of problems and drawbacks. In this section, we list some of the more relevant issues with emulation that should be considered and dealt with when applying emulation as a preservation strategy.

4.3.1. Legal issues

Emulation is impacted by legal constraints in two ways, namely copyright and license agreements. Leaving aside a small proportion of open source software, most software used in architectural practice that is nowadays finding its way to archives is subject to some form of copyright.[11] Although in some cases, open source alternatives may be available, they often don't support all features of a file format[12] and deviate from the original user experience and look & feel of the application.

The copyright barrier applies to all levels and types of software that is required to set up an emulation environment: from operating systems, to software libraries and applications. Eventually, some parts of an application may be copyrighted by other parties, such as software libraries, audio and video codecs or even fonts (Rosenthal, 2015). Copyright issues may be a reason to offer offline access only (e.g. in a reading room), but even when carried out by memory organisations and authorized under national laws, reproduction of computer programs and databases, is in conflict with EC Law (Anderson, 2011). The application of copyright restrictions can be enforced by legal actions, but also at a technical level, e.g. by Digital rights management (DRM).

Archival institutions may not be able to acquire software licenses because the vendor doesn't support or distribute the software anymore. License agreements may also be violated when software is made available in an emulated or virtual environment, as is the case with AutoCAD.[13] The current trend to replace perpetual software licenses with subscription licenses will only aggravate this problem in the future. Some of the applications and other copyrighted software may be orphan works (abandonware) e.g. when the vendor no longer exists or has been taken over by another company. Initiatives such as UNESCO's PERSIST project,[14] the Software Preservation Network,[15] the Society of American

[11] E.g. results of the 2016 survey mentioned in Leventhal, 2017

[12] E.g. LibreCAD for DWG files, https://librecad.org/

[13] http://download.autodesk.com/us/FY19/Suites/LSA/en-US/lsa.html

[14] https://unescopersist.org/

[15] https://www.softwarepreservationnetwork.org/

Archivists' Digital Design Records Task Force[16] have entered into relationships with software rights owners to overcome these legal barriers (Meyerson, 2016).

In contrast to a migration strategy (where files are usually normalized to an open file format, without copyright limitations and which typically can be opened in an open source software application), with emulation the uncertainty as to whether or not a software may be used legally will always be present. On the plus side, emulation does not require any modification of the original file. Therefore, copyright restrictions that prevent the modification of the original file (e.g. a non-derivative license[17]) do not apply (Hoeren, 2013).

It is clear that institutions will not provide access to emulated digital collections unless the legal basis for doing so is clarified (Rosenthal, 2015). In a scenario in which EaaS is offered by a third party provider, one can expect that this problem will be tackled by the service provider, and/or that this problem will be solved in a collaboration of the consortium of users. On the downside, a commercial service model on the other hand may be considered by the rightsholder as a commercial use, for which the a higher financial compensation will be claimed.

4.3.2. User experience and interaction levels

Fidelity to the original is an important reason for archival institutions to apply emulation as a preservation strategy. As Rosenthal (2015) states, just as migration doesn't provide perfect mimicry of the original user's experience, neither does emulation. On the one hand, emulation may well achieve quasi identical behaviour of a software application (going as far a slowing down the execution as to mimic the effect of less powerful CPU's). But changes in hardware (e.g. different keyboards and monitors resolutions, player controls such as joysticks) create different user experiences. Interaction with an emulated digital object (a file or an application) may also be hindered by the lack of appropriate connected hardware connections (e.g. obsolete SCSI interfaces, different keyboard layouts, joystick).

Finally – but most importantly – the question arises what kind of interaction with the original file a user will require: will it be sufficient to represent a file read only, or should a user be able to make changes to a file and eventually even save these changes? An answer to this is also an answer to the question whether it is sufficient to emulate software readers instead of fully functional software. If it is required that the entire environment is saved after a session finishes, this implies that multiple copies of the same disk images must be saved.

4.3.3. Software availability

As already mentioned, a range of software (operating systems, software libraries and applications) is needed to set up an emulation environment, many of which can be found in software archives on the web.[18] These archives operate in the twilight zone of what is legal and usually rely on voluntary work, so the long-term availability of these collections is not assured. An EaaS initiative can play a significant role in the provision of legacy software.

[16] https://www2.archivists.org/groups/design-records-section/digital-design-records-taskforce

[17] https://creativecommons.org/licenses/by-nc-nd/4.0/

[18] Examples are https://winworldpc.com/ and https://www.macintoshrepository.org/

However, even if software is distributed by the vendor is available, an emulation environment might need specific configurations, customizable plugins or scripts (e.g. Dynamo Visual Programming for Revit). Some of these software components can only be acquired from the actual owner of the software. In an EaaS strategy, organizations should have the possibility to have these applications included in the SaaS environment, and/or to manipulate the off the shelf software to comply with the original configuration.

4.3.4. Availability of emulation environments

All current emulation frameworks depend on the same set of open source emulators. Some of the most commonly used for preservation include:

- Basilisk II is an open source emulator of Macintosh computers with processors of the Motorola 680x0 family.[19] To install Basilisk II, a valid PPC Macintosh ROM image and a generic retail MacOS install CD or a disk image (.ISO) is required.[20] The ROM files are strictly speaking still copyright protected, although Apple seems to have stopped taking action against distribution of ROM files.[21]
- KEGS Kent's Emulated GS is an Apple IIgs emulator.[22]
- The Mini vMac emulator collection allows to run software made for early Macintosh computers sold from 1984 to 1996, also with Motorola's 680x0 processors. Mini vMac is part of the Gryphel Project.[23]
- PCE/Macplus is a classic Macintosh emulator.[24] It emulates a Macintosh 128K, Macintosh 512k, Macintosh 512ke, Macintosh Plus, Macintosh SE or a Macintosh Classic.
- Sheepshaver emulates a PowerPC Macintosh computer and is capable of running up to Mac OS 9.0.4. As with SheepShaver, Basilisk II requires a ROM and an MacOS install CD.[25]
- QEMU is a generic and open source machine emulator and virtualizer.[26] QEMU is developed as a tool to support software and hardware development and is used relatively little for preservation purposes. Thus, although it is a mainstream open source project under active development, it is difficult to get issues of concern for preservation addressed by the QEMU team (Rosenthal, 2015).
- DOSBox is an emulator program that emulates an IBM PC compatible computer running a DOS operating system.[27]
- MAME[28]: MAME (stood for Multiple Arcade Machine Emulator) absorbed the sister-project MESS (Multi Emulator Super System), so MAME now documents a variety of (mostly vintage) computers, video game consoles and calculators, in addition to the arcade video games that were its initial focus.

[19] https://en.wikipedia.org/wiki/Motorola_68000

[20] https://basilisk.cebix.net/

[21] https://www.emaculation.com/doku.php/ubuntu

[22] http://kegs.sourceforge.net/

[23] https://www.gryphel.com/c/minivmac/

[24] http://www.hampa.ch/pce/

[25] https://sheepshaver.cebix.net

[26] https://www.qemu.org/

[27] https://www.dosbox.com/

[28] https://www.mamedev.org/

2015). E.g. for each file type present, metadata must indicate the software and software version required to open the file. Software must be documented with requirements regarding the operating system, dependent software libraries and hardware (see figure 1).

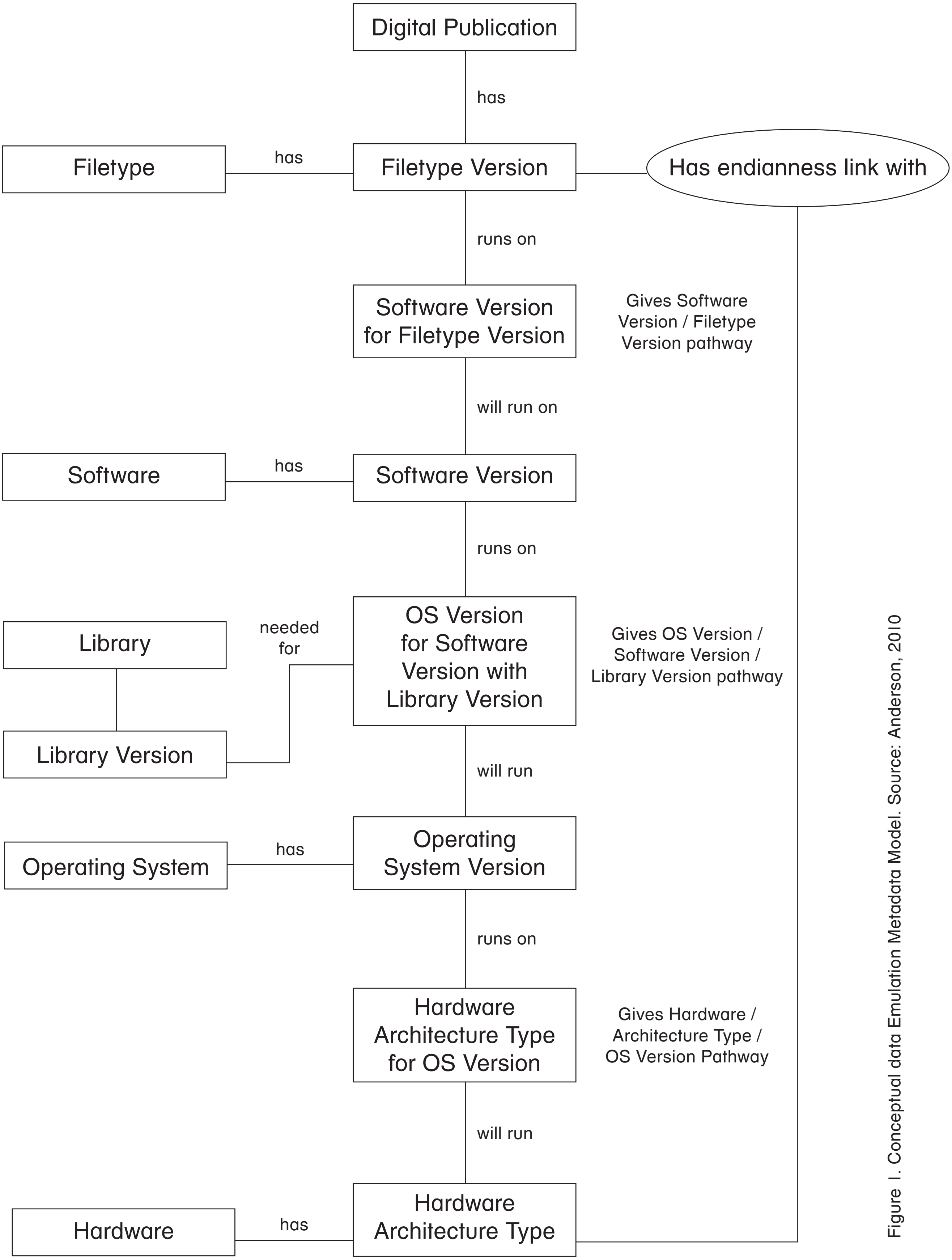

Figure 1. Conceptual data Emulation Metadata Model. Source: Anderson, 2010

Other emulators include 86Box[29] and Dioscuri (both x86 computer hardware emulators),[30] VICE (a Commodore Emulator),[31] UAE (Amiga emulator),[32] BeebEm (BBC Micro and Master 128 emulator),[33] JavaCPC (Amstrad CPC-Emulator),[34] etc. Not an emulator in the strict sense but worth mentioning is Rosetta: it was introduced on Mac OS X 10.4 for Intel-based Macs and allowed to run software for PPC-based hardware. Rosetta was discontinued by Apple after OS X 10.6.[35]

As the developers of GRATE-R in the PLANETS project testify, building, testing and maintaining emulators is only the first barrier to take when deploying an emulation infrastructure. The technical complexity of implementing an emulator requires in-depth skills in installing and configuring the emulator. Even then, every single file to be emulated requires a complex procedure (von Suchodoletz, 2010a).

Most emulators are made available under an open source license, with many of them relying on donations and volunteering to provide the necessary updates and support. This may raise concerns about the sustainability of emulation software. Indeed, in a fast changing technical environment, an emulator itself can be exposed to technical obsolescence (Rechert, 2016). Similarly, most of the efforts within the digital preservation community regarding research on emulation and the development of emulation tools, are project based and lack long term sustainability.

The KEEP approached the problem of obsolete emulators with the concept of emulator migration, with the aim of developing a virtual machine as a platform for emulators. This virtual machine is designed to be migratable without difficulty to any conceivable hardware platform on the future (Anderson, 2010). Disk images must be kept under the same conditions as any other file in the digital archive, including technology watch. Maintaining disk images is actually similar to a 'classic' digital object migration strategy. The properties to watch for technical obsolescence are the technical interfaces between software used and (external) hardware components (Rechert, 2016). If an external dependency becomes unavailable, the disk image (and the software environment it contains) or the artefact itself must be adapted so they can perform again in the changed technical environment. Better cooperation and sharing of resources through the creation of an ecosystem for emulation tools and services on a more structural basis is required to address these threats to the sustainability of the emulation strategy. An EaaS solution fits very well in this more cooperative approach.

4.3.5. Metadata

In order for an emulation work process to run smoothly, it is important that all components are clearly documented. The exact format (including the version) of each file must be documented extensively and in detail. (Rechert e.a.,

[29] https://86box.github.io/

[30] http://dioscuri.sourceforge.net/ Latest update was in 2011.

[31] https://vice-emu.sourceforge.io/

[32] https://amiga.technology/

[33] http://www.mkw.me.uk/beebem/

[34] http://www.cpcwiki.eu/index.php/JavaCPC

[35] https://www.apple.com/rosetta/index.html

In a highly automated emulation environment, these metadata must be available in a structured and standardized form so that the required components are accurately selected. Ideally, these metadata should be recorded in a standard format such as PREMIS, but as yet there are no broadly adopted standards for describing the technical, provenancial, and relational properties of software (Meyerson, 2019). Metadata provided by tools such as Siegfried[36] or DROID[37], both based on the PRONOM[38] database, are not adequate yet for the needs of emulation. The tools cannot identify or record the dependencies that specify the emulator, operating system and application needed to emulate a preserved artefact (Rosenthal, 2015). Recently, efforts have been made to promote the use of Wikidata as a central resource for software and file format metadata (Thornton e.a., 2017).

4.3.6. Cost considerations

Calculating the cost of emulation and comparing this with a migration strategy is a complicated task, and many different factors have to be taken into account.[39] Emulation might be the cheaper option, because the task of normalization to an open file format is not necessary once a reliable emulation environment is available. On the other hand, the complexities of an emulation environment are far higher than the relative straightforward migration approach, and will therefore probably require more resources.

The introduction of an emulation strategy along with migration, may also involve the implementation of two technologies and work processes that compete for available resources in an organization.

As with technological issues, cooperation between organizations can be a means to divide efforts and thus reduce costs related to personnel and hardware. Adoption of standards, which is a clear prerequisite in cooperative models, is likely to make emulation more feasible and cost effective (Granger, 2000).

4.3.7. Local, private or public cloud

When implementing emulation, institutions can decide to provide this service locally (e.g. in a reading room or an exhibition environment) or remote. A choice for local access may be motivated by technical considerations, e.g. when an emulation environment is provided as an ad hoc solution for a specific collection or file format. If a more user-friendly service is desired and/or multiple emulation environments are offered, this is a good reason to develop emulation in a private or public cloud environment. A private cloud has the advantage that the institution is in full control, which may be appropriate when dealing with proprietary software or confidential files. As mentioned before, a public cloud solution has the benefit that institutions do not need to maintain their own infrastructure. When emulation services for different original environments are made available in a unified and standardized way, economies of scale might result from this (von Suchodoletz, Rechert and Valizada, 2013). However, sharing emulation infrastructure via the

36 https://www.itforarchivists.com/siegfried

37 https://github.com/digital-preservation/droid

38 https://www.nationalarchives.gov.uk/PRONOM

39 E.g. see Bøgvad, 2014

cloud may arise new difficulties. User access must be strictly regulated and properly secured. Functionalities such as data exchange from removable media must be abstracted from the emulators and made available remotely.

4.4. Implementing emulation in a preservation environment

To summarize the points listed above we can say that emulation can be a viable approach in a digital preservation strategy, especially when it comes to complex digital objects or when a user experience close to the original environment is important. Emulation as a preservation strategy, however, presents new challenges, such as higher technical complexity and legal obstacles. Emulation in a public cloud (as a service), where resources such as expertise, infrastructure and personnel are shared, can be part of the solution. Nevertheless, a thorough knowledge of the software environment is necessary for the archival institution itself.

In order to build a successful preservation strategy and practice based on emulation (either as a service or maintained locally), an archival institution will need to arrange a number of matters and will have to develop certain procedures.

4.4.1. Define target audience and usage

An important question is what level of interaction with the emulated environment will be supported: will a user only be able to view the digital object, or will he/she be able to transform the object? If transformation is allowed, will a user be able to save the result and download it? Or should a user be able to save the transformed object within the emulation environment?

The question is related to the usage and the audience that is envisaged with the emulation service. The target audience can vary between staff, researchers or a wide audience. For staff, usage requirements could be limited to migrating files to an open file format (i.e. as part of a migration strategy). An exhibition audience will want to be able to interact with the object, while researchers may want to effectively transform the files. They even may want to be able to upload a file themselves, manipulate and transform it and download it again. This requires the emulator to be able to connect to input/output devices of the underlying hardware. The level of expertise of the audience may also be a factor that determines the desired level of usability. Ideally, a user should have the ability to find a file in any given (obsolete) format, click on a link to the file and have it 'automagically' (Cochrane, 2018) open it in an environment that is identical or resembles the native software stack.

A related question is the extent to which the 'look & feel' of the original environment must be reflected by the emulation environment. E.g. if it is desirable to simulated the screen resolution or even the screen technology (CRT vs. LCD) of the original hardware, then measures must be taken to adapt the hardware according to those requirements.

4.4.2. Document the software environment requirements of digital objects.

The appropriate software environment for the digital object and application needs to be determined, in particular its required operating system, other system settings and dependencies, technical interfaces to the hardware layer (e.g. drivers for input/output) and hardware requirements (e.g. memory requirements). This can be based on a reference installation (i.e. software already installed and configured on a existing – probably legacy – computer system). All this information must be stored in a structured manner, and managed with the same care as the other preservation metadata of the digital objects in the archive.

Documentation and metadata creation should also be applied to the disk images that are generated or created. Although most disk image formats support the embedding of metadata to some degree,[40] it is advisable to maintain the disk image metadata in the same preservation system as the one used for the digital objects in the archive.

4.4.3. Acquire software and licenses

Emulation environments will need legacy software, and although a service can provide some of the more mainstream software packages, it will most probably also rely on a user community to acquire obsolete software packages to be included in the service. These packages can be found in its original environment (e.g. the architect's computer) or from a software archive such a VetusWare,[41] the Internet Archive[42], WinWorld[43] or the Macintosh repository.[44] If the applications are taken from an external repository, a license must often also be acquired in order to use the software (legally).

The necessary software layers and components will most often be assembled into a disk image, containing all necessary components to read and transform the digitale object. These disk images can also be created directly from the original hardware (e.g. an architect's computer). If this is not available, the environment must be constructed through the installation of necessary software components. Rechert (2016) recommends to use both approaches.

In this context, it is important to look beyond the horizon of desktop applications: the tendency of professional applications to be available on mobile devices (smartphones, tablets) might proliferate in the architectural profession maybe faster than we expect. Whereas we currently deal with a spectrum of three operating system families (Windows, MacOS and Linux), this might evolve to a plenitude of systems, which will be even harder to manage and monitor. Moreover, the tendency to provide software as a service for day to day software applications (Office 365, Google Suite, Adobe cloud, …) will sooner or later also impact the CAD software market. In a world where cloud based applications are the norm, acquiring a software application becomes almost impossible.

[40] E.g. the Open Virtualization Format (OVF), http://www.dmtf.org/standards/ovf

[41] http://vetusware.com/

[42] https://archive.org/details/software

[43] https://winworldpc.com/

[44] https://www.macintoshrepository.org/

4.4.4. Define a legal policy

An archive institution that makes parts of its collection available to users must develop an adequate legal policy. In the first place this concerns the copyrights of the archive creator. Agreements about this are usually made when the archive is acquired. When digital objects are made available in an emulation environment in the public cloud, an adjustment to the agreement with the archive creator is likely to be required. Second, the archival institution must determine how it will deal with risks related to the use of software whose copyrights are not clear (eg abandonware) or whose license agreement may be violated. This often comes down to making a risk estimate. E.g. as mentioned before, Apple seems to have stopped taking action against distribution of ROM files, which are needed to run a predecessor of macOS in an emulator such as SheepShaver or Basilisk II – but in the end this remains illegal.

4.4.5. Set-up, maintenance and technology watch

An EaaS solution will usually be built in line with an existing preservation environment, which may require additional components to establish the integration between the preservation solution and the emulation environment. The level of integration is closely related to the level of automation. Cochrane (2017) distinguishes three levels of automation for matching emulatable software environments with digital objects:

- In an automated workflow, the user selects an object to interact with. The file formats and creation dates of the digital object are determined to find the environment containing the most compatible software.
- With the semi-automated solution, a staff member confirms that the recommended environment is actually a good match for the files stored on the object.
- In the manual scenario a staff member manually selects and configures an environment for each object.

Since the purpose of an emulation strategy is that the file and its environment is preserved and maintained unaltered, digital objects and (native or compatible) software theoretically need no maintenance other than the bit level preservation. At every other level, the necessary precautions must be taken to monitor threats from technological obsolescence. In a service model, these tasks may be executed by the service provider – but this does not discharge the archive institution from the responsibility to monitor this service and to respond adequately if it does not meet the requirements.

5. EaaS at HNI

5.1. Context

5.1.1. HNI's preservation policy

The preservation policy of HNI is defined in a formal document. The preservation policy defines [digital] preservation as the "Recording, storing, managing and making available digital documents (in the broad sense of the word) in such a way that they can be consulted, can be accessed and remain authentic over time." (Ras, 2018). In this policy, HNI follows the OAIS model[45] as a guideline for setting up the infrastructure and necessary processes. Besides OAIS, the policy document mentions a number of standards, a.o. the *ISO 13008:2102 Information and documentation – Digital records conversion and migration process and the NEN-ISO 23081 Standaard voor metadata*.

Among the responsibilities of the organization, the HNI preservation policy mentions:

- The submission agreement, which includes agreements about access rights and the actions HNI is allowed to perform on the digital objects. Preferably, the archive creator also provides a detailed description of the context and structure of the supplied archive. This description might include a description of the software environment.
- Arrange intellectual property rights, by which the archive creator gives permission to the HNI to carry out actions to keep and preserve the digital information objects in a sustainable manner. The policy doesn't yet mention rights settlement between right owners other than the archive creator.
- Together with the archive creator, HNI determines the designated community (i.e. the intended and future users) of the digital information objects. In practiceboth researchers as well as a broader audience have access to the archive.
- HNI ensures that users are able to understand and use the information that is made available.

The preservation functionality of HNI is based on bit preservation. This means that acquired digital archives will at least be preserved at bit level. In addition, carefully selected parts of these archives will be managed at a higher level. This functional preservation requires more than just bitstream preservation. A preservation plan defines the principles behind this, and the eventual interventions when the state of information objects is threatened.

At pre-ingest, a risk assessment of any format present in the collection will be made to determine how sustainable the format is. Based on this, normalization of certain formats to a more sustainable format may follow. Both formats – including the metadata – are preserved. The access functionality supports the authorization and provision of a DIP via a link to the collection management system and the access workflow. Depending on the designated community or user, the information can be made available in various ways, for example via

[45] ISO 14721 Space data and information transfer systems – Open archival information system (OAIS)

a viewer or as a download. HNI envisages the drafting of a file format policy document as a completion of the preservation policy (Ras, 2018).

The basic principles of HNI's preservation policy seem primarily to focus on a migration strategy, but do not rule out a different approach. HNI also participates in a software preservation research project (Neggers, 2020), which is consistent with the required actions outlined above when a migration strategy is to be followed.

5.1.2. HNI preservation infrastructure

HNI's preservation infrastructure is built around the Archivematica software. Normalization is Archivematica's primary format preservation strategy. In accordance to this strategy, preservation copies are added to the AIP and the access copies are used to generate a DIP for upload to the access system. Note that the original files are always kept, to allow for different preservation actions in the future, such as emulation.[46] The implementation of Archivematica doesn't rule out the application of an emulation strategy, but neither does it provide enough functionality to support emulation. As a minimal requirement, the HNI preservation environment should be able to provide the original file as part of the DIP in order to make this file manually available for emulation.

5.1.3. HNI born digital collection profile

HNI manages approximately 700 archives and collections of Dutch architects, urban planners, professional associations and study programs with a total of 4,000,000 documents. The archive of so-called 'born digital' material is growing rapidly and now has 60,000 files.[47] HNI investigated the problems related to the preservation of born digital documents on the basis of the archive of the architectural firm MVRDV. This archive contains 400 projects from the period 1993–2015, which together contain 5 TB of data in many types of formats: in the design process, various software programs are used and experimented with. The MVRDV archive contains a number of files in formats such as DWG,[48] DXF,[49] QuarkXPress,[50] MiniCad[51] and 3D Studio,[52] which are are eligible for emulation. A second archive that is used as a test cases is the archive of landscape architect Michael van Gessel, containing born digital files such as e-mails and DWG files.

5.2. Implementation in preservation policy, workflows and infrastructure

File format policies specify (1) which file formats are accepted in a repository, and (2) what measures are taken for each file format to ensure preservation and accessibility of the files in this format. In many cases, the policy defines an archival format to which the files will be normalized (e.g. a MS Word document

[46] https://www.archivematica.org/en/docs/archivematica-1.11/user-manual/preservation/ preservation-planning/

[47] https://collectie.hetnieuweinstituut.nl/

[48] https://www.nationalarchives.gov.uk/pronom/fmt/531

[49] https://www.nationalarchives.gov.uk/pronom/fmt/63

[50] https://www.nationalarchives.gov.uk/pronom/x-fmt/182

[51] https://www.nationalarchives.gov.uk/pronom/x-fmt/1136

[52] https://www.nationalarchives.gov.uk/pronom/x-fmt/19

will be migrated to a PDF/A file). The decision as to whether emulation is the preferred strategy for a given file (format) depends on an assessment that involves multiple parameters, as mentioned in the issues and implementation tasks above:

– Definition of users and usage requirements (e.g. exhibition setting, migration support, user interaction and transformation).
– Availability of software components in the desired emulation environment (applications, software libraries, operating systems and emulators) and software licenses).
– Suitability of the emulator and other components (e.g. support of necessary hardware components such as USB drive, keyboard, monitor, sound card etc).
– Legal admissibility of every component in the emulation environment with regard to its intended use.
– Availability of alternative solutions (e.g. software alternatives that run on contemporary operating systems, virtualization environments) and their costs.
– Risk analysis with regard to the sustainability of the components in the emulation environment.

When an emulation strategy is adopted (either as a replacement or complementary to a migration strategy), the file format policy should be updated with relevant information answering these issues. Regardless of specific formats, an institution may also decide to archive the entire working environment of an archive creator (e.g. an architect's pc) as a disk image. A practical implementation will also affect some workflows and the software architecture of the digital archive:

– Stipulation in the submission agreement of the admission of HNI to emulate digital objects through third party service.
– Collection and machine-readable storage of metadata related to all components in the emulation environment.
– Connection of the emulation environment with the digital archive management system (i.e. Archivematica). In an optimal scenario, selecting a file or disk image in Archivematica should trigger the emulation environment automatically.

Building, maintaining and using emulation software and services requires expertise on multiple levels, such as being familiar with native software and the file formats, the software environment and the emulator. Even if the emulation application is provided as a service, a thorough knowledge of the other two levels is still required.

5.3. Example implementation scenarios

As an illustration, we consider three formats present in the MVRDV and van Gessel archives which are possibly eligible for emulation, namely DWG, QuarkXPress and MiniCAD. As target audiences and user needs are not yet specified in the preservation policy, we assume that files in these formats are accessible for both researchers and a broader audience. In the following chapters we investigate the rationale for emulation of these file formats, the technical feasibility, legal obstacles and possible alternatives.

5.3.1. AutoCAD DWG

DWG is the native file format of the proprietary AutoCAD software, but has evolved to a de facto standard format for CAD in architecture and engineering (Vanstappen, 2020). First released in the 1980's, the format and software has a long history of increasingly new functionalities. AutoCAD DWG is backwards compatible, meaning files created in any release can be opened and edited in the same or any later release. As the DWG format is backwards compatible, there is no immediate need for emulation or virtualization to open, edit or convert a DWG file. However, it can be interesting to view and manipulate a file in the software that originally created and edited it, to gain a better understanding of the design process. As second reason is that DWG files can have relationships with multiple dependent files of different types, which represent parts of the digital object (other .dwg files, textures, plot files, etc.) DWG files are therefore difficult to isolate from their original environment, and emulation of a fully archived environment can reduce the risk of misrepresentation of the digital object.

AutoCAD runs on an x86 architecture. Subsequent versions run on the DOS or any of the Windows operating systems, but some have been available for the Macintosh operating system 7, running on a Motorola 68K processor. Images of the installation disks, including serial numbers can be found on the web.[53] System requirements are poorly documented, but we found version R11 (1988) can run perfectly on DOS 6.22 with as little as 4 MB RAM (Miller, 1991).

The x86 DOS versions of AutoCAD (DOS or Windows) can be emulated with DOSBox or virtualized with Virtualbox. Virtualbox can also be used to recreate an environment for later Windows versions. For the Macintosh version of AutoCAD, SheepShaver or Basilisk II is a more complex, but feasible option.

AutoCAD is proprietary software, so it is mandatory for an institution to own a license to the software in order to use it. The copyright owner AutoDesk does not seem to be bothered by the fact that outdated versions of the software are freely available on the web, but no indications were found that Autodesk formally renounces any claim on copyrighted material or will accept making license keys available to third parties. As mentioned earlier, AutoDesk's license agreement doesn't allow the virtualization of it's software in a networked environment – which is a complicating factor when providing public or private cloud based access to the software. Although there do not seem to be many technical or practical barriers to virtualizing or emulating AutoCAD, it is legally hardly feasible to provide a risk-free emulation / virtualization of AutoCAD in a public or private cloud environment.

An alternative can be found in the use of free, open source alternatives, such as LibreCAD.[54] However, these open source alternatives do not support all AutoCAD features and may therefore may give a distorted image of the original document.

[53] E.g. https://winworldpc.com/product/autodesk-autocad/r11-mac or http://vetusware.com/

[54] https://wiki.librecad.org/

Another alternative is the user of web enabled viewers such as DWG Fastview (relying on the Flash technology),[55] or the Web CAD SDK from cadsofttools.[56]

5.3.2. MiniCAD/VectorWorks

VectorWorks is a CAD program, originally called MiniCAD. MiniCAD was released for the Macintosh operating system in 1985, while the first Windows version didn't appear until 1996. The specification of the VectorWorks (.vwx) or MiniCAD (.mxd) format has not been released, nor is it available via reengineering as is the case with DWG files. The MiniCAD file format is not backward compatible. Opening older files (before MiniCAD 6) even requires a two-stage migration, with two different versions of VectorWorks. Each of these versions run on different operating systems and hardware requirements. Earlier versions of MiniCAD can only run on earlier Macintosh operating systems (up to system 9) with Motorola 68K or PowerPC based hardware.

The lack of backwards compatibility can be a good reason to set up an emulation environment, even if it's only to allow for a migration strategy. Just as DWG, MiniCAD/VectorWorks has evolved significantly over the years, so emulation can be a way for a better understanding of the design process.

The software required to open and access earlier MiniCAD files could be obtained from older hardware or from software archives on the web.[57] Depending on the emulated operating system, this can be achieved with Mini vMac, Basilisk II or SheepShaver.

Emulating the Macintosh operating system encounters legal hurdles. As mentioned above, both an original operating system and a ROM file are required to obtain a working emulator. In particular, the legality of using a ROM file is still unclear – even when it was obtained from existing hardware. Unlike DWG, the file formats of MiniCAD / VectorWorks have not reached the de facto default status, leaving little or no alternative software available to read these files.

5.3.3. QuarkXpress

QuarkXPress is a desktop publishing software for creating and editing page layouts that can be run on Mac and Windows operating systems. It was first released in 1987. Versions 1-2 were available on Macintosh only.[58] With the release of QuarkXPress 3.0 in 1990, Quark quickly achieved a dominant position in the desktop publishing market and became the standard for desktop publishing. By the end of the 1990's they gathered a market share of around 90%.[59]

The QuarkXPress proprietary file format is not backwards compatible. E.g. QuarkXPress 9 is the last version that will open legacy documents from QuarkXPress 3, 4, 5 and 6. For migration purposes, it can be necessary to set up an emulation environment. Like AutoCAD and MiniCAD, this software has

[55] https://en.dwgfastview.com/

[56] https://cadsofttools.com/products/web-cad-sdk/; https://beta.sharecad.org/

[57] E.g. https://www.macintoshrepository.org/, where we found a copy of MiniCAD

[58] https://en.wikipedia.org/wiki/QuarkXPress

[59] https://en.wikipedia.org/wiki/Quark_(company)

evolved greatly, which can be an argument for recreating the user experience of previous versions through emulation.

Working versions of the application are available from software archives on the web.[60] Since version 1992 an application running on x86 hardware is available, so virtualization may be a viable option. Another alternative is to get hold of a copy of the QuarkXPress Document Converter. This is a standalone application which converts legacy documents (QuarkXPress 3, 4, 5 and 6) to the new format (QuarkXPress 9.1) enabling users to open these documents in QuarkXPress 10 or later without having QuarkXPress 9 or older installed. The converter was available for free from QuarkXpress.[61]

In legal terms, the same problems occur as with the file formats discussed above: both the required operating systems and the software itself are in principle still subject to copyright.

[60] E.g. https://winworldpc.com/product/quarkxpress/1x

[61] https://support.quark.com/en/support/solutions/articles/19000055856-how-can-i-open-legacy-documents-created-in-quarkxpress-version-7-or-below-in-quarkxpress-2016-

6. Conclusion and recommendations

For the HNI's born digital collection, emulation is an option to consider for several reasons: taking into account the complexity of CAD files, a migration strategy is not entirely appropriate to meet the wishes of HNI's target audience. Emulation is more capable of displaying digital objects in their original environment and realizing an authentic experience. In some cases, this strategy is even the only way to make these objects accessible. As software and software platforms evolve, more file formats will become obsolete and emulation may sooner or later be the only possible strategy.

The knowledge and technology to make digital objects (files or full disk copies) accessible in an emulation environment is available, but often still fragmented and developed in temporary projects or by organizations whose sustainability can be questioned. As Rosenthal (2015) points out, the two most important barriers are that the tools for creating preserved system images are inadequate, and that the legal basis for delivering emulations is unclear, and where it is clear it is highly restrictive. A service model in which public cloud emulation environments are offered to multiple organizations can realize economies of scale and thereby achieve a more sustainable ecosystem.

It is therefore an attractive option for HNI to be part of a – yet to be realized – partnership and community that develops this service model and makes it available to the archiving community. In that case too, however, HNI will have to resolve a number of issues for itself. It will have to decide whether and how emulation will be offered to end users: will an emulation environment be offered for each file format, or will the choice be made to offer complete environments in the form of a disk image based on the original hardware? Or is emulation only used as part of a migration strategy? These choices will have to be part of the global preservation policy and elaborated in the file format strategy.

On a technical level, this will have to be translated into requirements with regard to the digital repository management system, in particular the way in which this is linked to the emulation environment, and the way in which required metadata to control this process are registered. It will have to be decided what level of automation will be supported.

The procedures and work processes must also be adapted. For example, it is important to document the technical environment in detail when acquiring archives. This moment can be an opportunity to acquire software, document and archive. This software can then be hosted in an emulation environment, and potentially shared with other collaborative partners.

Finally, HNI must formulate a policy with regard to the legal feasibility and possible risks of acquiring, using and offering software that is still copyrighted.

7. Glossary

AIP (Archival Information Package)
An information package containing both the archived digital object and the metadata that describes its structure and content. It may consist of multiple files and multiple representations of the original digital object.

Born digital
Refers to materials that originate in a digital form, in contrast to digitazation, through which analog materials become digital.

Digital object
A computer file or combination of files containing a logical unit of information.

Bit level preservation
A very basic level of preservation of the digital object as it was submitted (literally preserving the bits forming a digital object).

DIP (Dissemination Information Package)
An information package, derived from one or more AIPs, received by the consumer in response to a request.

Disk image
A computer file containing the contents and structure of a disk volume or of an entire data storage device, such as a hard disk drive.

DRM (Digital rights management)
Technologies for restricting the use, modification, and distribution of copyrighted digital works (such as software and multimedia content).

Emulation
A technique for implementing a virtual machine on a host computer whose instruction set is different from the host computer's.

Emulator
The host computer software that executes the translation between host computer and the emulated environment.

File format
A standardized way that information is encoded for storage on a digital storage medium.

File format policy
An official statement of preference for specific file formats over others, sometimes expressed as a recommendation, other times as a set of requirements for deposit into a digital repository. The policy can include preservation measures an institution will take for each given format.

Functional preservation (aka content preservation or logical preservation)
Preservation actions that ensure the continued accessibility of of the content of digital objects over time, which may include the generation of new technical versions of the objects.

Preserved system image
A set of stored data that is input to an emulator or a VMM and thereby executed. The set of data would normally include metadata describing the VM's configuration, the contents of its storage media, and optionally the state of the virtual machine's memory. A preserved system image might be the result of imaging a physical computer's disks or an image created by installing from vendor's install media for the operating system and applications needed, together with preserved files requiring that environment.

Public cloud
Computing services offered by third-party providers over the Internet, either free or sold on-demand. Services may include storage capabilities, applications or virtual machines. Public cloud allows for scalability and resource sharing that would not otherwise be possible for a single organization to achieve.

Runtime environment
The configuration of hardware and system software. It includes the CPU type, operating system and any runtime engines or system software required by a particular category of applications.

Significant properties
Aspects of a digital record that must be preserved over time in order for it to remain accessible and meaningful.

Virtual machine (VM)
a computer that has no separate physical existence, but is part of the behavior of a physical computer, called the VM's host computer. VMs mimic the instruction set and hardware configuration of some physical machine, or an abstract machine such as the Java Virtual Machine.

Virtualization
A technique for implementing a VM on a host computer. It depends on the host computer's instruction set being the same as (strictly mostly a superset of) the VM's instruction set, and having certain specific hardware properties that enable virtualization. Almost all instructions executed by the VM are directly executed by the host computer's CPU; only a few instructions are intercepted, using the specific properties, and performed by host software. Virtualized systems run unmodified software binaries.

8. Resources

Aerts, P. (2016). Software Sustainability, Final Report. Den Haag: NCDD

Allen, E., & Daily, C. (2018). Reassessing the Principles of Preservation in the Wake of Born-Digital Materials. (Retrieved from https://dx.doi.org/10.2139/ssrn.3173958)

Anderson, D., Anderson, J. & Pinchbeck, D. (201) Toward A Workable Emulation-Based Preservation Strategy: Rationale and Technical Metadata. In: New Review of Information Networking 15(2):110-131

Anderson, D. (2011). A layman's guide to the KEEP legal studies. European Commission. (Retrieved from https://researchportal.port.ac.uk/portal/files/150295/D2.6_laymansguide legalstudies_final.pdf)

Bøgvad Kejser U., Hougaard Edsen Johansen K., Thirifays A., Bo Nielsen A., Wang D., Strodl S., Miksa T., Davidson J., McCann P., Krupp J., and Tjalsma H., 2014, D3.1 Evaluation of Cost Models and Needs & Gaps Analysis Final Report (Revision 1), (Retrieved from http://www.4cproject.eu/d3-1/)

Butler, B., Aufderheide, P., Jaszi, P. A., & Cox, K. L. (2019). Cracking the Copyright Dilemma in Software Preservation: Protecting Digital Culture Through Fair Use Consensus. Butler, B., Aufderheide, P., Jaszi, P., and Cox, K.," Cracking the Copyright Dilemma in Software Preservation: Protecting Digital Culture Through Fair Use Consensus," Journal of Copyright in Education and Librarianship, 3(3).

Cochrane, E. (2014). Emulation as a Service (EaaS) at Yale University Library. The Signal. (Retrieved from https://blogs.loc.gov/thesignal/2014/08/emulation-as-a-service-eaas-at-yale-university-library/)

Cochrane, E., Tilbury, J. & Stobbe, O. (2017) Adding Emulation Functionality to Existing Digital Preservation Infrastructure.iPres Conference 2017. (Retrieved from: https://ipres2017.jp/wp-content/uploads/45Euan-Cochrane.pdf)

Cochrane, E. (2018). Designing a Universal Virtual Interactor (UVI) for digital objects. (Retrieved from https://www.dpconline.org/blog/idpd/designing-a-uvi-for-digital-objects)

Cochrane, E., Tilbury, J., & Stobbe, O. (2018). Adding emulation functionality to existing digital preservation infrastructure. Journal of Digital Media Management, 6(3), 255-264.

De Vos, J. (2020) Software archivering en de Nederlandse Erfgoedsector. Netwerk Digitaal Erfgoed. (Retrieved from https://www.netwerkdigitaalerfgoed.nl/wp-content/uploads/2020/04/NDE_ Rapport-Software-archivering.pdf)

Granger, Stewart (2000) Emulation as a Digital Preservation Strategy. In: D-Lib Magazine 6(10). (Retrieved from https://doi.org/10.1045/october2000-granger)

Hoeren, T., Kolany-Raiser, B., Yankova, S. & Hecheltjen, M. (2013) Legal Aspects of Digital Preservation. Edward Elgar Publishing
Holdsworth, D., Wheatley, P.R.: Emulation, Preservation and

Abstraction. In: RLG Diginews, vol. 5 (4), (2001).

Koninklijke Bibliotheek (n.d.) What is emulation? (Retrieved from https://www.kb.nl/en/organisation/research-expertise/research-on-digitisation-and-digital-preservation)

Leventhal, Aliza (217) CAD/BIM Taskforce Report. (Retrieved from https://www2.archivists.org/sites/all/files/CAD%20BIM%20Taskforce%20Report%202017.pdf)

McDonough, J., Olendorf, R., Kirschenbaum, M., Kraus, K., Reside, D., Donahue, R., Phelps, A., Egert, C., Lowood, H., & Rojo, S. (2010). Preserving Virtual Worlds Final Report

Meyerson, J., Vowell, Z., Rosenthal, D., Cochrane, E., & Milic-Frayling, N. (2016) Software Sustainability And Preservation: Implications For Long-term Access To Digital Heritage. (Retrieved from https://unescopersist.files.wordpress.com/2016/05/iprespanel_submission- cameraready.pdf)

Meyerson, J. (2019) 3D/VR Preservation: Drawing on a Common Agenda for Collective Impact, In: Jennifer Grayburn, Zack Lischer-Katz, Kristina Golubiewski-Davis, and Veronica Ikeshoji-Orlati (eds). 3D/VR in the Academic Library: Emerging Practices and Trends. Council on Library and Information Resources.

Miller, A.R. (1991) The ABC's of AutoCAD release 11. Sybex

Molenda, A. (2018) Between Creators and Keepers. How HNI Builds its Digital Archive. (Retrieved from https://collectie.hetnieuweinstituut.nl/sites/default/files/between_creators_and_keepers_final.pdf)

More, P. (2014) Emulation or virtualization: What's the difference? Retrieved from: https://blog.dell.com/en-us/emulation-or-virtualization-what-s-the-difference/

Neggers, F. (2020) Inrichting Digitaal Archief. Het Nieuwe Instituut. (Retrieved from: https://collectie.hetnieuweinstituut.nl/bewaren/inrichting-digitaal-archief)

Ras, M. (2018) Preservation policy. Het Nieuwe Instituut. Versie 1.0 (Retrieved from https://collectie.hetnieuweinstituut.nl/sites/default/files/preservation_policy_erfgoed_nl29nov.pdf)

Rechert, K., Falcao, F. & Ensom, T. (2016) Introduction to an emulation-based preservation strategy for software-based artworks.

Rechert, K., Liebetraut, T., Stobbe, O., Valizada, I., & Steinke, T. (2015, November). Characterization of CD-ROMs for Emulation-Based Access. In iPRES (p. 144). (Retrieved from http://ir.las.ac.cn/bitstream/12502/8542/1/ipres2015-proceedings.pdf)

Rosenthal, D. S. (2015). Emulation and virtualization as preservation strategies. (Retrieved from https://mellon.org/media/filer_public/0c/3e/0c3eee7d-4166-4ba6-a767-6b42e6a1c 2a7/rosenthal-emulation-2015.pdf)

Rosenthal, D.S. (2019). Emulation as a Service. (Retrieved from https://blog.dshr.org/2019/08/emulation-as-service.html)

Roy, J. (2019). Preserving interactivity: towards next generation digital preservation philosophy and systems. (Retrieved from https://mspace.lib.umanitoba.ca/handle/1993/34406)

Thornton, K., Cochrane, E., Ledoux, T., Caron, B., & Wilson, C. (2017, September). Modeling the domain of digital preservation in Wikidata. In Proceedings of ACM International Conference on Digital Preservation, Kyoto, Japan. (Retrieved from https://ipres-conference.org/ipres17/ipres2017.jp/wp-content/uploads/7.pdf)

van der Hoeven, J., van Wijngaarden, H., Verdegem, R, & Slats, J. (2005) Emulation – a viable preservation strategy. (Retrieved from https://www.kb.nl/sites/default/files/docs/Emulation_research_KB_NA_2005.pdf)

Vanstappen, H. (2019) A preservation policy for the AutoCAD DWG/DXF file format. HNI/Datable (Retrieved from http://datable.be/files/2019_HNI_RapportDWG-DXF_v1-0.pdf)

von Suchodoletz, D., Rechert, K. & Valizada, I. (2013). Towards Emulation-as-a-Service: Cloud Services for Versatile Digital Object Access. In: International Journal Of Digital Curation vol. 8(1)

von Suchodoletz, D., Rechert, K., Schroder, J., van der Hoeven, J., Roberts, B., & van den Dobbelsteen, M. (2010a) Manual on the use of emulation services as part of a preservation strategy. (Retrieved from https://www.planets-project.eu/ManualOnEmulationServices.pdf)

von Suchodoletz, D., van der Hoeven, J. R., Rechert, K., & Schröder, J. (2010b). Seven steps for reliable emulation strategies: solved problems and open issues. In Rauber e.a. 7th International Conference on Preservation of Digital Objects (iPRES2010). Computer Society.

Walsh, T. (2018). Efficient appraisal and processing of disk images of legacy digital storage media at the Canadian Centre for Architecture. In: Journal of Digital Media Management, 6(4), 357-369.

Welch, I., Rehfeld, N., Cochrane, E., & von Suchodoletz, D. (2012). A practical approach to system preservation workflows. PIK-Praxis der Informationsverarbeitung und Kommunikation, 35(4), 269-280. (Retrieved from https://www.researchgate.net/publication/235956241_A_Practical_Approach_to_System_Preservation_Workflows)