



Project Number	IST-2006-033789
Project Title	Planets
Title of Deliverable	Report describing results case studies
Deliverable Number	D3
Contributing Sub-project and Work-package	PA/5
Deliverable Dissemination Level	External No restrictions
Deliverable Nature	Report
Contractual Delivery Date	31 st May 2008 (M24)
Actual Delivery Date	30 th May 2008 (M24)
Author(s)	ALUF, BL, IBM, KB-NL and NANETH

Abstract

This deliverable is meant as a first exploration of the possibilities, strengths and weaknesses of the application of emulation and virtualization as a preservation approach.

Keyword list

Case study, emulation, virtualization, experiment

Contributors

Person	Role	Partner	Contribution
Remco Verdegem	PA/5 lead	NANETH	Document owner and co-author
Peter Bright	member PA/5	BL	Co-author
Paul Wheatley	member PA/5	BL	Co-author
Jasper Schröder	member PA/5	IBM	Co-author
Jeffrey van der Hoeven	member PA/5	KB-NL	Co-author
Dirk von Suchodoletz	member PA/5	ALUF	Co-author

Document Approval

Person	Role	Partner
Remco Verdegem	WP lead PA/5	NANETH
Frank Houtman	Subproject lead PA	KB-NL
Bjarne Andersen	SB member	SB

References

Ref.	Document	Date	Details and Version
FRA	Framework and Workflow for testing tools for different technical environments	M13	Internal deliverable, PA/5-D1, issue 3.

EXECUTIVE SUMMARY

This deliverable considers the use of emulation (tools) and virtualization (tools) in the context of digital preservation. The use of emulation and virtualization in the context of digital preservation is an area that needs more detailed exploration. Previous work has scratched the surface of this problem, but does not provide all the answers. It is for this reason that the partners of PA/5 decided to conduct a number of case studies. The purpose of the first iteration of these case studies is to explore the use of emulation/virtualization as a preservation approach and to perform experiments using digital content from real life collections provided by the Planets partners. A variety of digital documents – websites, text documents, geospatial data, computer games and spreadsheets – and a variety of emulation and virtualization tools – Dioscuri, VirtualBox, Virtual PC, QEMU, VMware, MESS and UVC - are used in these case studies. The results are promising: emulation and virtualization have proven to be a feasible preservation approach. As PA/5 we realize that there are still issues to be resolved, like setting up a so called 'software archive' and finding ways for performing 'remote emulation', in order to provide emulation a more solid basis within the digital preservation community. Another issue to be solved is what we called the presence of 'implicit knowledge', being knowledge you will not find in manuals or other documentation, but that is merely present in the heads of people (experience). How to record and preserve this knowledge is an interesting question.

In the second iteration of the case studies the focus will be on experimenting with existing emulators within a remote emulation framework. Also, in-depth research questions will be addressed defining what is required to setup a remote emulation framework (PA/5-D9, month 36).

TABLE OF CONTENTS

1.	Introduction	5
2.	Case study overview	6
2.1	KB-NL: Experiencing an old website	6
2.2	NANETH: Rendering obsolete WordPerfect files	6
2.3	BL: Legacy Map Software	6
2.4	ALUF: Execution of different computer games	7
2.5	IBM: preserving interactivity	7
3.	Results of the case studies	8
3.1	Case study 1: Experiencing an old website	8
3.1.1	Background	8
3.1.2	Define Basic Experiment Properties	9
3.1.3	Design Experiment	9
3.1.4	Specify Resources	9
3.1.5	Run Experiment	10
3.1.6	Evaluate Experiment	12
3.2	Case study 2: Rendering obsolete WordPerfect files	14
3.2.1	Background	14
3.2.2	Define Basic Experiment Properties	14
3.2.3	Design Experiment	14
3.2.4	Specify Resources	15
3.2.5	Run experiment	17
3.2.6	Evaluate Experiment	19
3.3	Case study 3: Legacy Map Software	21
3.3.1	Background	21
3.3.2	Define Basic Experiment Properties	21
3.3.3	Design Experiment	21
3.3.4	Specify Resources	21
3.3.5	Run Experiment	22
3.3.6	Evaluate Experiment	23
3.4	Case study 4: Execution of Different Computer Games	24
3.4.1	Background	24
3.4.2	Define Basic Experiment Properties	24
3.4.3	Design Experiment	25
3.4.4	Specify Resources	25
3.4.5	Run Experiment	26
3.4.6	Evaluate Experiment	26
3.5	Case study 5: Preserving interactivity with digital objects	29
3.5.1	Define Basic Experiment Properties	29
3.5.2	Design Experiment	29
3.5.3	Specify Resources	29
3.5.4	Run Experiment	29
3.5.5	Observations	30
3.5.6	Evaluate Experiment	31
4.	Conclusions and lessons learned	32
4.1	General	32
4.2	Lessons learned	32
4.3	Further research	33
4.3.1	Disk image preparation	33
4.3.2	Software preservation	33
4.3.3	Remote emulation	33

1. Introduction

The use of emulation in the context of digital preservation is an area that needs more detailed exploration. Previous work has scratched the surface of this problem, but does not provide all the answers. PA/5-D1 "Framework and Workflow for testing tools for different technical environments" (May 2007) provides a starting point for the development of successful approaches to emulation testing, but the next logical phase has to be practical experience and experimentation.

It is for this reason that the partners of PA/5 decided to conduct a number of case studies. The purpose of these case studies is to explore the use of emulation as a preservation approach and to perform experiments using digital content from real life collections provided by the Planets partners. While Planets must be realistic in what it can achieve with limited resources, the design and execution of emulation experiments on sample content running under emulation will enable significant progress to be made.

The use of case studies enables us to increase our understanding of the use of emulation and virtualization for digital preservation purposes and will provide us with insights into the key aspects of emulation or virtualization. For instance, what is really required to actually apply emulation as a preservation strategy? Is it sufficient to give the customer/researcher access to the emulation tool itself analogous to a migration tool or are other, extra steps required?

Accordingly, the focus of our case studies is not limited to merely executing experiments to test the accuracy of specific emulation tools (e.g. Dioscuri¹, QEMU, and MESS) and virtualization² tools (e.g. Virtual PC and VirtualBox), but also to considering all aspects of the use and application of emulation and virtualization within the context of digital preservation.

¹ Dioscuri was a joint project of KB-NL and NANETH (2005-2007) with the aim to develop a modular emulator in order to prove that emulation is a viable preservation approach. After finishing the project (May 2007) Dioscuri was adopted by PA/5 for further development under the umbrella of Planets.

² "... virtualization software does not emulate an entire computer with all (peripheral) components, but instead only certain (usually privileged) operations. The virtualisation software arbitrates access to the underlying hardware, generally allowing direct access where safe, and providing software-emulated access otherwise. But it remains dependent on the specific underlying computer architecture. That is why virtualisation software can not run on any kind of platform. Instead, emulation is much more flexible because it mimics the exact functionality of all computer components without any ties to specific host platforms. Virtualisation is often faster than emulation, but it is less flexible as it is very tight to the underlying computer system." (PA/5-D1 Framework and Workflow for testing tools for different technical environments; FRA).

2. Case study overview

2.1 KB-NL: Experiencing an old website

Although websites are assumed to be platform independent they still rely on hardware and software. To preserve long-term access to a website, all its dependencies such as a web browser, fonts, operating system, and plug-ins should be considered. As these environmental aspects change over time via new releases of browsers, new font sets, new operating systems or new versions of plug-ins, so too will the look-and-feel of a website.

Emulation offers the possibility to authentically render a computer environment (hardware and software) and can be used to render websites from any time. This case study is concerned with rendering an old website using Planets' Dioscuri emulator and will identify the requirements and capabilities of using emulation for accessing web content.

In this, the following research questions are considered:

- What is needed to render a website via emulation?
- What problems are encountered?
- What aspects of emulation or emulation tool require further research?

2.2 NANETH: Rendering obsolete WordPerfect files

Over the course of the years the interface between the word-processing software and the end user has undergone a drastic change. At the beginning of the 1980's documents were produced using a simple non-graphical interface. WordPerfect 4.2 is a well-known example of this early type of applications. Commands were given by using a range of specific key combinations. These were displayed in what was referred to as the "underwater screen". The combination of the document's content, appearance and structure was only visible upon printing the document to paper (and in "Print Preview").

The objective of this case study is to investigate the rendering of obsolete WordPerfect files using Dioscuri. Samples of real life WordPerfect files are selected to be included in these experiments.

The following research questions are considered:

- What is required to actually use an emulation tool for rendering obsolete WordPerfect files in a digital preservation context?
- What steps have to be taken?
- What problems are encountered?
- What are lessons learned?
- What aspects of emulation require further research in the next iteration?

2.3 BL: Legacy Map Software

The BL's Map department receives many items that combine "data" (which may be images, geospatial databases, maps, etc.) with "programs" (to provide a front-end to that data). The software programs are for many of the items an essential part of the experience; for example, the data may be stored in proprietary formats, or the application might provide ways of locating and using the images. Typically these are Windows programs, but many of them are quite old and suffer incompatibility issues with recent versions of Windows. As such, there is a desire to provide suitable emulation environments to allow these programs to be installed and their data used.

Two specific programs were examined for the purposes of the case study, the *Atlas Schoemaker* and *Panaïrama*.

The *Atlas Schoemaker* contains digitized images of the 18th century historical-topological atlas assembled by Andries Schoemaker, and a Windows front-end to search and view those images. Although it would be clearly desirable to compare the virtual machines with "real" hardware running the software, this was unfortunately not possible due to a lack of such hardware; instead, best guess appraisals were made according to our own experience and expectations.

Panairama contains digitized aerial photographs of the city of Perth, Australia, along with a front-end for viewing the images.

The research questions of the experiment are:

- Does current commodity virtualization software allow the use of legacy PC operating systems to enable access to the maps software described above?
- How much expertise is necessary to successfully set up and install the virtual machines?

2.4 ALUF: Execution of different computer games

Computer games are part of our digital heritage, and so in many countries national libraries have collected them. A strategy is therefore needed to preserve these objects in such a way as to give access to them.

Computer games make good tests of the characteristics of emulation software because they often make extensive use of the computer's hardware (especially audio and video output and user input), as well as being performance-sensitive. For the first round of tests some popular games of the 1980s and 1990s were selected:

- "Arkanoid" (produced by Taito for the Commodore 64 in 1986)
- "Loom" and "Indiana Jones and the Fate of Atlantis" (LucasArts, 80386 PC, 1990 and 1992)
- "DOOM" (first release, id Software, 80386 PC, 1993)
- "Myst" (Brøderbund, 80386 PC, 1996)
- "Command & Conquer": Red Alert (Westwood Studios, second release, 80486 PC, 1996).

The research questions of the experiment are:

- What is needed to actually run an ancient computer game?
- Is the rendering complete?
- What kind of documentation is needed?
- What are the strengths and weaknesses of emulation in general and the chosen emulation tool in particular (in this case MESS/VMware/QEMU)?
- What are the lessons learned?
- What aspects of emulation or emulation tool require further research in the next iteration?

2.5 IBM: preserving interactivity

The Universal Virtual Computer is a specification of a simple computer with a limited instruction set, designed with the assumption that a new emulator for this computer can be developed on future hardware. We have developed emulators for the UVC in both Java and C++. It takes about 1 to 2 person months to create an emulator. Previous work showed how format interpreters written for the UVC are used to convert binary objects (like GIF, JPEG) to a human readable/understandable format (text). As the format is understandable, now and in the future a restore program can be developed that recreates the presentation of the object.

This previous research handled only static features of the preserved objects and did not involve general interaction with the outside world. In this case study, interaction with a preserved digital object is developed and tested.

The research question of this case study is:

- How to preserve the interaction with a digital object using the UVC?

3. Results of the case studies

3.1 Case study 1: Experiencing an old website

3.1.1 Background

In 2003, the average lifetime of a website was about 44 days³. Today, it is expected that this has not changed much. Based on this it can be concluded that information found on the internet is very fleeting. While this source for information plays a growing role of importance in the process of doing research, sources of today may become inaccessible tomorrow. To cope with this problem, various national initiatives are currently undertaken to preserve at least some of this online information. For example, the National Library of Australia has been running the PANDORA programme for twelve years now, to periodically capture websites in the .au domain. In the US, the Internet Archive has captured website all across the world since 1999. In the Netherlands, the KB-NL also started to archive websites⁴. The KB-NL's archiving is done on an opt-in basis, meaning that website owners are requested to give their permission before the KB-NL starts harvesting and archiving their website for the long term. Currently, around 1000 websites have been harvested this way and many will follow in the next years.

Although websites are assumed to be platform independent they still rely on hardware and software. To preserve long-term access to a website, all its dependencies such as a web browser, fonts, operating system and plug-ins should be considered. As these environmental aspects change over time via new releases of browsers, new font sets or new operating systems, so too will the look-and-feel of a website. For example, a missing Adobe Flash plug-ins prevents you from watching online YouTube movies. Browsers themselves also show variations; many people will still remember the differences in rendering a website with Netscape Navigator compared to the same website in Internet Explorer. These notable differences are all caused by the underlying configuration of the computer environment. Therefore, it is essential not only to preserve website content, but also to retain access to the authentic environment that renders it.

Emulation offers the possibility to authentically render a computer environment (hardware and software) and can potentially be used to render websites from any time. This case study is concerned with rendering an old website using Planets' Dioscuri⁵ emulator and will identify the requirements and capabilities of using emulation for accessing web content.

3.1.1.1 The case

A scientist in digital media is doing research into the evolution of the web in the Netherlands in the early period of the internet: 1994-2000. He is especially interested in the look and feel of websites around that period.

As the KB has a large collection of Dutch websites archived in their e-Depot, the electronic archive of the KB, he decides to visit the reading rooms of the KB. There, he browses through the catalogue which files all website titles (and URLs) available in the collection. Although digital files in the collection can be retrieved automatically online, access to archived websites is still under construction. For now, requested websites have to be setup manually before a requester can experience it on a computer in the reading rooms of the KB-NL. In general, this process works as follows:

1. The visitor selects a website from the online catalogue of the KB-NL.
2. An assistant of the public hall is notified about the request.
3. The requested digital object (website) is disseminated from the e-Depot.
4. After dissemination, the object is going through a Planets post-process:
 - a. A characterization tool analyses the files of the website.
 - b. Because the website is old no current rendering solutions exist anymore. Therefore, the Planets suite offers several alternatives to access the website using a selected

³ L. Martin, "Where websites go to die", <http://www.smh.com.au/articles/2003/10/16/1065917549444.html>

⁴ http://www.kb.nl/hrd/dd/dd_projecten/webarchivering/index-en.html

⁵ <http://dioscuri.sourceforge.net/>

preservation action (migration/emulation). Because the scientist is interested in the look-and-feel emulation is chosen as action.

- c. The tool registry offers a best practices emulator that can render the original environment of the website.
- d. The emulation environment is prepared and started.
- e. The emulation process renders the website on the computer of the scientist.

3.1.2 Define Basic Experiment Properties

The goal of this experiment is to identify the requirements and capabilities of using emulation for accessing web content. In this, the following research questions are considered:

What is needed to render a website via emulation? Which problems are encountered? What are the strengths and weaknesses of emulation in general and the chosen emulation tool in particular (in this case Dioscuri)? What aspects of emulation or emulation tool require further research?

3.1.3 Design Experiment

The objective is to render an old website from the period between 1994 and 2000 using the original software and Dioscuri as replacement for hardware. This experiment is an example of "Emulation of obsolete platform in order to render a digital object. This might include using an interactive multimedia application or running a word processor in order to view a document", as defined in PA/5-D1 section 6 - Emulation uses.

The strategy is to setup a computer environment from around 1994-2000. This period spans the lifetime of operating systems from MS-DOS 5.0 to MS Windows 2000. As the current version (February 2008) of Dioscuri (0.3.0) can run 16-bit operating systems stably, MS-DOS is chosen; modern versions of the Windows operating system require a 32-bit instruction processor which is currently only experimentally supported by Dioscuri. On top of the operating system, a browser is required. As internet around 1994 was still in its infancy, few web browsers for MS-DOS were available⁶. However, one browser that originates from 1996 is still extant and freely available: Arachne⁷. Arachne is very compatible with early operating systems and currently the most advanced graphical web browser for MS-DOS-based computers. For the experiment Arachne 1.60 has been used.

Because the KB-NL has only recently started its web archiving program (2006), no websites have been preserved from before 2006. However, as the KB-NL has a website of its own, the communications department was able to deliver the first graphical KB-NL website from its private collection. This website originates from 1997 and was first available on the domain www.konbib.nl (as domains consisting of two characters were not allowed in those days).

3.1.3.1 Limitations

A crucial limitation is that Dioscuri does not yet support pointing devices such as a mouse. Therefore, only a keyboard can be used to navigate the website.

3.1.4 Specify Resources

3.1.4.1 Host platform

To run the emulation process, a host computer platform is required. For this experiment an HP Workstation with Intel Pentium 4 3.8 GHz with 2 GB RAM has been used. It runs MS Windows XP Professional SP 2. Furthermore, it has a preinstalled Sun Java Runtime Environment (JRE) 1.6.0 for running Dioscuri (a Java application).

3.1.4.2 Target platform

The target hardware platform is specified as follows:

1. 16-bit x86-based CPU
2. 1 MB RAM (expandable)
3. 10 MB HDD

⁶ http://en.wikipedia.org/wiki/List_of_web_browsers

⁷ http://en.wikipedia.org/wiki/Arachne_%28web_browser%29

4. XT/AT/PS2 compatible keyboard
5. virtual screen
6. VGA video graphics adapter
7. DMA-support
8. IRQ-handling based on a Intel 8259 PIC
9. Timing mechanism based on an Intel 82C54 PIT and Crystal Clock
10. Real-time clock with integrated CMOS
11. RS232 Serial port based on UART 16550A
12. System BIOS using Plex86/Bochs BIOS
13. Video BIOS using VGA LGPI'ed BIOS

For running target software, a hard disk image has to be created that needs to be attached to Dioscuri. To create a hard disk image, the simple image manipulation tool WinImage for Windows is used. WinImage is capable of creating FAT-16 images with Master Boot Record (MBR). Furthermore, the image can contain any file that is desired by easily dragging and dropping files from the host computer into the disk image.

The disk image compiled for this experiment consists of a preinstalled version of MS-DOS 5.0.0 and the DOS web browser Arachne 1.60. Also, the KB-NL website has been included as a local copy on the disk because Dioscuri does not support network access.

[3.1.4.3 Configuring the emulator](#)

The final step in the preparation process is to configure the emulator. The following parameters were set:

- a. CPU set to 5 MHz
- b. RAM set to 1 MB
- c. Floppy disk support was disabled
- d. Hard disk support was enabled and the disk image has been attached as bootable hard disk.
- e. Boot sequence was set to boot from HDD first

All other values were left as default.

[3.1.5 Run Experiment](#)

After starting the Dioscuri application a GUI is presented which offers the user to start the emulation process. Starting this process first shows the typical black screen booting the computer via the familiar boot process. After a minute, the MS-DOS prompt comes on screen.

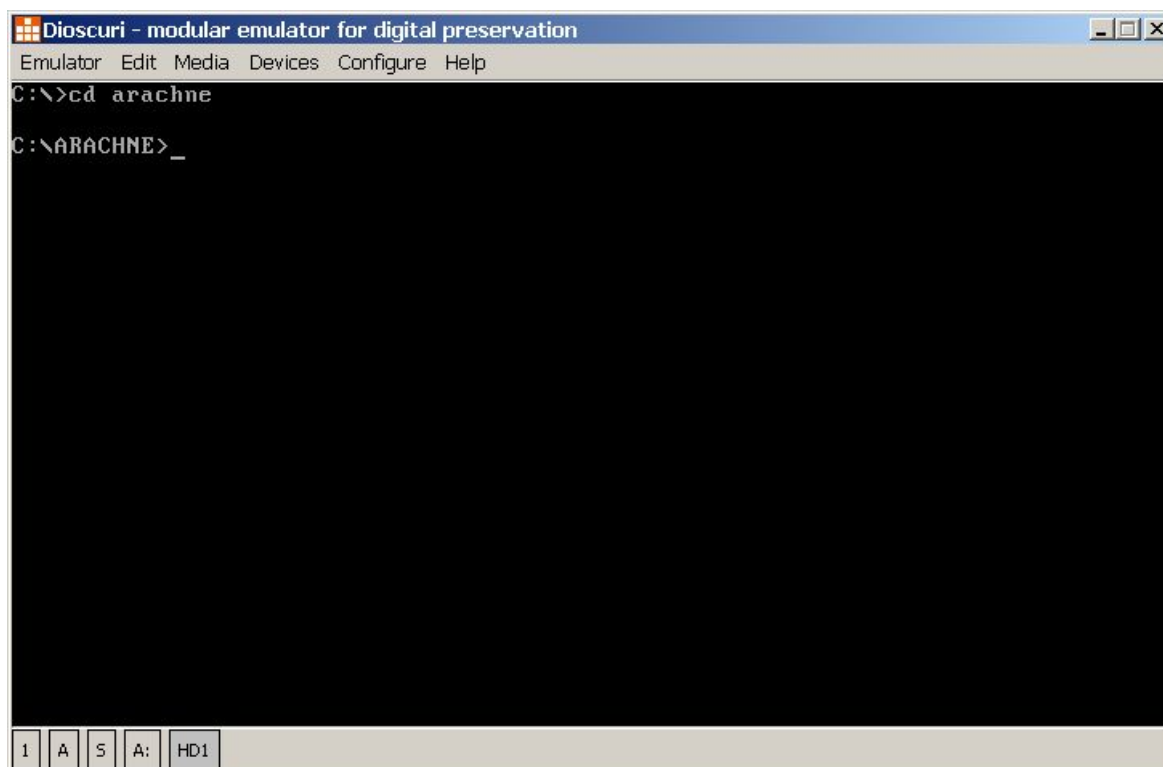


Figure 1: Dioscuro running MS-DOS 5.0.0

To open the website, first Arachne has to be started.

Typing: <cd\arachne arachne> starts the web browser. The screen changes from text mode into graphics mode. Within Arachne, the user has to navigate via the arrow keys of the keyboard to move the graphical cursor over the screen. Selecting the browse option allows the user to browse the file system on the hard disk. The folder has to be selected containing the locally stored web site. After that, the index.html has to be chosen.

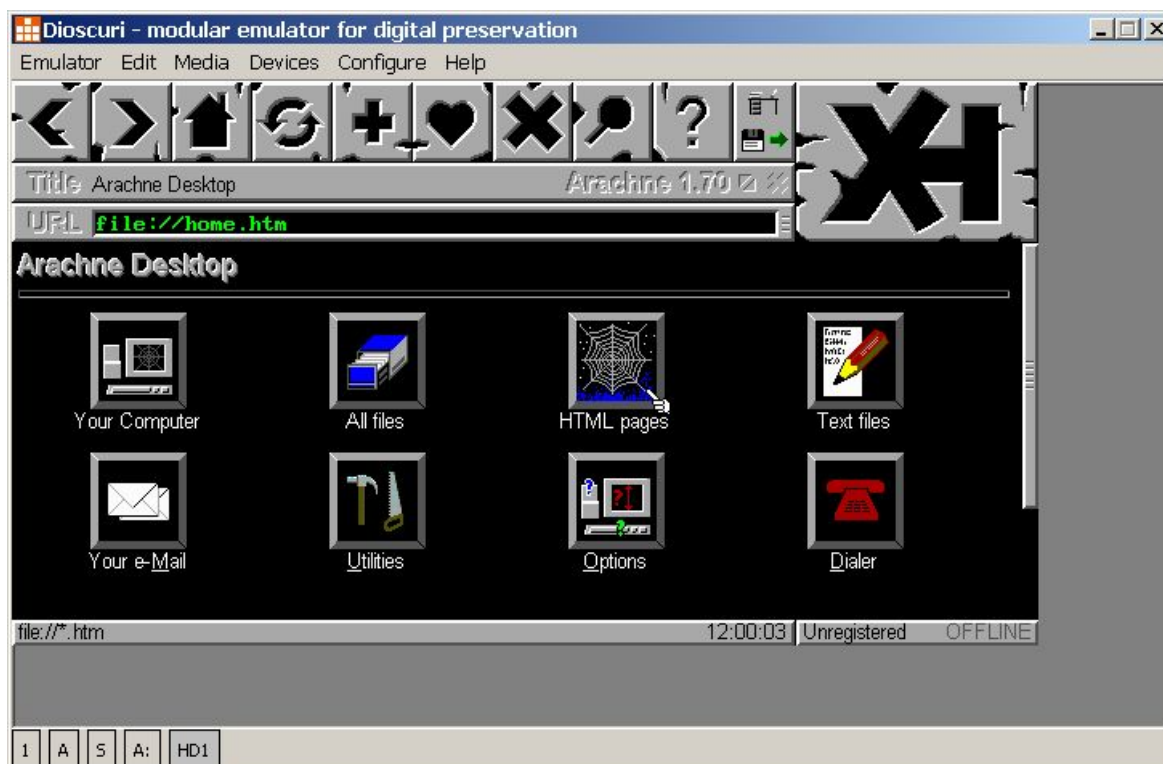


Figure 2: Dioscuro running Arachne web browser

Next, the web site is loaded on screen, ready for the user to navigate through it and compare it with the screenshot of the original website.



Figure 3: DioscURI rendering KB website

3.1.6 Evaluate Experiment

Referring to the research questions mentioned in the beginning of this experiment, the following things were noted:

3.1.6.1 [What is needed to render a website via emulation?](#)

In this experiment a minimum set of requirements were used. Obviously, with a DOS-emulator, Arachne website and a disk imaging tool the original web site can be rendered again. However, it would be more interesting to have a web site stored on a server and access it via a virtual network card under emulation on a client computer.

3.1.6.2 [What problems are encountered?](#)

3.1.6.2.1 Absence of original environment

Unfortunately, no original hardware platform could be found for comparing the emulated environment with its physical equal. Instead, a screenshot of the original website in its authentic appearance has been used to compare it with the emulated rendition. Although this limits the criteria that can be used for visual comparison of both environments (behaviour can not be verified accurately) it already gives a good impression of what the website looked like and how users must have been navigating through it.

3.1.6.2.2 Unexpected issues

First of all, the web site did not open as expected. Comparing the screenshot of the original web site with the emulated one showed significant differences. After a short investigation, it showed that Arachne does not handle JavaScript very well which happens to be applied in the original web site. To get the web site working, some adjustments were made to the original object by removing some JavaScript elements from the HTML code. After these adjustments, the web site turned out to open and render almost correctly with Arachne. Some difference was noted in background colour of the website, but all other aspects seemed to work correct. The different renderings of the same website in different browsers demonstrate a widespread problem with web sites: different browsers will show the same page in different ways.

3.1.6.2.3 Emulator

Starting the emulation process went without problems. However, typing text at the DOS-prompt showed some delay between typing the keystroke and actually appearing on screen. Although this is not a big problem, it might be confusing as the user may be unsure of which character he already has typed.

3.1.6.3 [What aspects of emulation or emulation tool require further research?](#)

It is worthwhile investigating more web sites and to include other web browsers. Another interesting experiment would be to have network support under emulation.

3.2 Case study 2: Rendering obsolete WordPerfect files

3.2.1 Background

Word processors play an important role in the creation of government documents. In the years since personal computers first appeared on the market at the beginning of the 1970's, word processing has grown into the most frequent office-automation application.

Over the course of the years the interface between the word-processing software and the end user has undergone a drastic change. At the beginning of the 1980's documents were produced using a simple non-graphical interface. WordPerfect 4.2 is a well-known example of these early types of applications. Commands were given by using a range of specific key combinations. These were displayed in what was referred to as the 'underwater screen'. The combination of the document's content, appearance and structure was only visible upon printing the document to paper (and in 'Print Preview'). Modern software packages employ WYSIWYG (What You See Is What You Get) and GUIs (Graphical User Interface) for the user's convenience. In addition, modern word-processing software incorporates a continuously-increasing range of advanced features. However, this means that printing the document to paper no longer guarantees that the document can be viewed and used in the manner that was originally intended.

3.2.1.1 The case

The Nationaal Archief of the Netherlands has potentially a large number of floppy disks (3,5 inch and 5,25 inch) stored in the depots for paper records. These media were transferred to the Nationaal Archief in the process of transferring paper records. As these media are stored within paper dossiers or case files, it is not clear where these media are located physically. The Nationaal Archief has the plan to start a project this year in order to find out how many digital records are stored on electronic media in our depots for paper records. The first step would be to actually find these media between the paper records. The Nationaal Archief has about 90 km of paper records! After finishing this inventory, the next step would be trying to render the content of these floppy disks. "While it is now generally understood that digital information must be copied to new storage media quite frequently, since such media become obsolete in a few years, there is a deeper problem that must be solved as well. Digital information can only be rendered usable and meaningfully by running the appropriate software and such software - along with the hardware in which it runs - can become obsolete just as quickly as the media on which the information is stored. The software that must be run to make a given digital document usable must understand the 'logical format' of the document, which is what enables the document to be made intelligible. Without appropriate software, trying to read a document is like trying to read hieroglyphics without the Rosetta Stone.⁸". The question to answer is whether or not these media still contain any accessible content. There is a chance that some media have become inaccessible, due to circumstances not optimal for storing electronic media. If part of these media have survived time, it is crucial to establish the file format of the files on the media. That information is important in order to install the right version of the software (the operating system and the application software) needed to 'authentically' render the digital records. If the original platform (hardware, operating system and application software) is no longer available, emulation can be considered. In this case study it is assumed that these media in our depots for paper records contain WordPerfect files. Furthermore it is assumed that the original hardware is no longer available and that an emulator is used, more specifically Dioscuri, to render these WordPerfect files.

3.2.2 Define Basic Experiment Properties

Research questions:

What is needed to actually use an emulation tool for rendering obsolete WordPerfect files in a digital preservation context? What steps have to be taken? What problems are encountered (and were not foreseen initially)? What are the lessons learned? What aspects of emulation or emulation tool require further research in the next iteration?

3.2.3 Design Experiment

The objective is to render obsolete WordPerfect files on 5,25 and 3,5 inch floppy disks, using Dioscuri. A few samples WordPerfect files are selected to be included in these experiments. These files were created by employees of the Nationaal Archief of the Netherlands around 1995 when

⁸ Jeff Rothenberg, "Using Emulation to Preserve Digital Documents", <http://www.kb.nl/pr/publ/usingemulation.pdf>

WordPerfect 5.1 was the leading word processor application, at least in the Netherlands. Dioscuri will be downloaded and installed (version 0.3.0., 16 bit version). On top of the emulator MS DOS 5.0 will be running as Operating System and WordPerfect 5.1 as end user application. While working on these experiments it appeared that one of the employees of the Nationaal Archief had 'preserved' a working copy of the Compaq Contura 430C laptop, which would enable us to compare the results of rendering WordPerfect files using Dioscuri with an obsolete hardware and software environment (486DX/4 100 MHZ processor with 8 MB RAM; running Windows 3.1 and WordPerfect 5.1).

3.2.4 Specify Resources

3.2.4.1 Preparation

To run the emulation process, a host computer platform is required. The experiments were executed on a laptop, HP nc8230 with Windows XP Professional, version 5.1, Service Pack 2. 2 GB RAM, Intel Pentium processor, 1,86 GHz. Furthermore, it has a preinstalled Sun Java Runtime Environment (JRE) 1.6.0 for running Dioscuri (a Java application).

The first step was to download and install Dioscuri. Dioscuri is an x86 computer hardware emulator written in Java. It is designed to ensure that digital objects and programs from the past can be accessed now and in the future. Dioscuri has two key features: it is durable and it is modular. Because it is implemented in Java, it can be ported to any computer platform which supports the Java Virtual Machine (JVM) without any extra effort. This reduces the risk that emulation will fail to work on a single architecture in the future, as it will continue to work on another architecture. As Dioscuri is completely component-based it also is flexible. Each hardware component is emulated by a software surrogate called a module. Combining several modules allows the user to configure any computer system, as long as these modules are compatible. New or upgraded modules can be added to the software library, giving the emulator the capability to run different target platforms.

The target hardware platform is a 16-bit X86-based CPU with MS DOS version 5.0 or higher and WordPerfect version 4.2 or 5.1.

3.2.4.2 Construct images

Dioscuri only supports virtual media, which means it does not give access to physical storage devices. Instead, a disk image has to be created from the original physical carrier (floppy or hard disk). This can be done using image creation tools, like WinImage. Dioscuri currently supports two types of virtual carriers: floppy disks and hard disks.

a. Floppy disk

A floppy disk image should be an uncompressed linear binary file (flat file) and should not be larger than 1.44 Megabytes.

b. Hard disk

Hard disks can be of any size (although the file system will limit the amount of available space). Just like floppy images, hard disk images have to be uncompressed linear binary files (flat files).

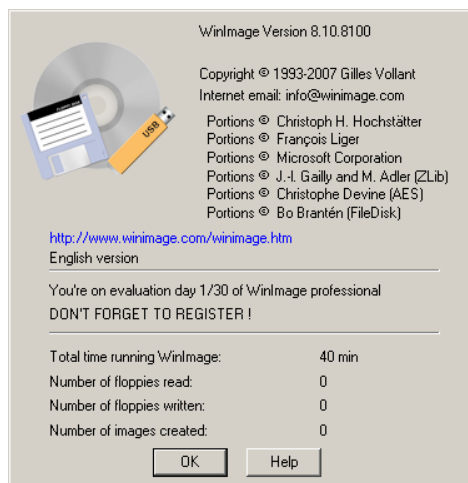


Figure 4: WinImage version 8.10.8100

3.2.4.3 [Create image WordPerfect files](#)

Instead of creating a separate floppy disk image of WordPerfect files it was decided to add the specific WordPerfect files to an existing image, already containing the required operating system (MS DOS 5.0) and application software (WordPerfect 5.1). At this moment it is not possible to attach two different hard disks at the same time to Dioscuri.

3.2.4.4 [Configure Dioscuri](#)

After having downloaded and installed Dioscuri, the emulator had to be configured, making sure that the right hard disk image is attached to Dioscuri. This image (MSDOS5_wp51_nl_remco.img) consists of MS DOS 5.0, WordPerfect 5.1 and several WordPerfect files.

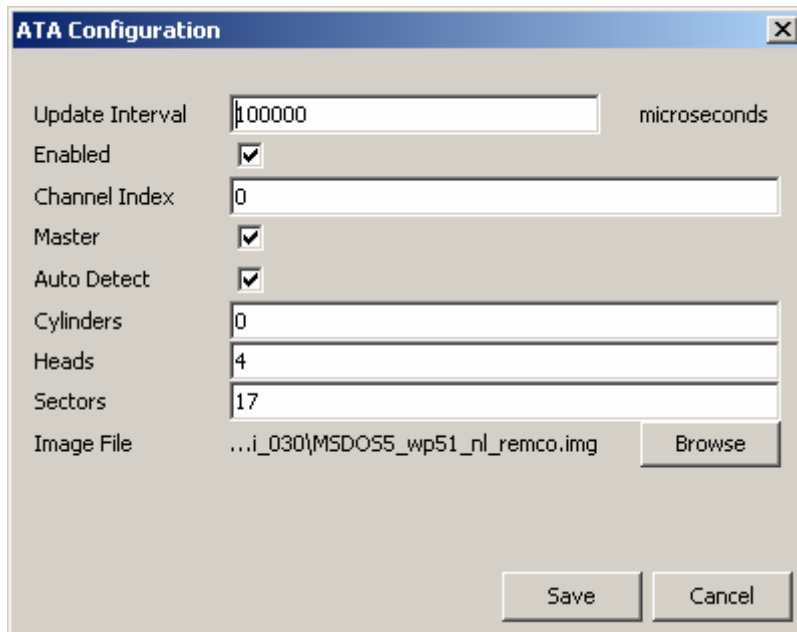


Figure 5: Configuring hard disk

Apart from mounting the right image, it was important to make sure that the correct version of the CPU was selected, being the 16 bits version of Dioscuri at a speed of 5MHz (see image)

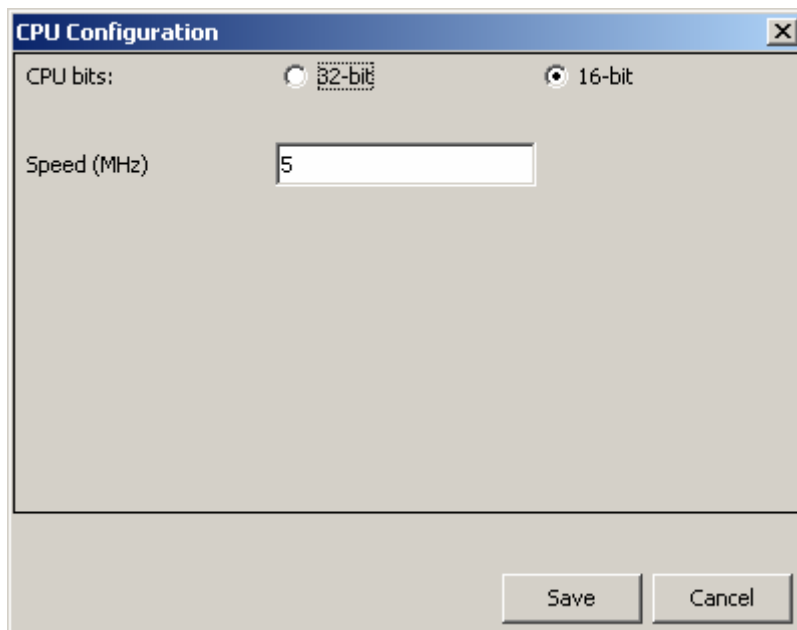


Figure 6: Configuring processor

After having saved all the changes to the configuration file, Dioscuri was started without any problem.

3.2.5 Run experiment

Having started Dioscuri a DOS box is presented to the user. Using the command line the correct directory (C:\wp51) is selected and WordPerfect is started by typing in the name of the executable file <wp>, invoking wp.exe in the mentioned directory.

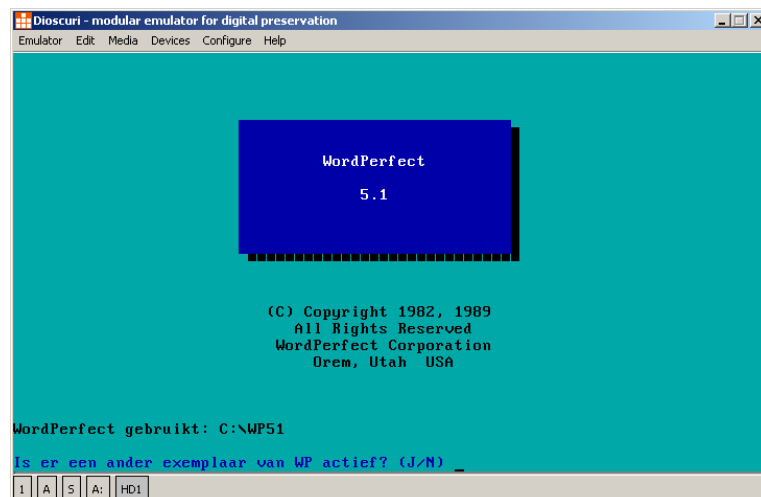


Figure 7: Start-up screen WordPect 5.1

The first major problem occurred: which function key, keystroke or combination of keystrokes was needed to open a WordPerfect file or to present the list of files comparable to <Open File> in MS Word. After a process of trial and error it was found that <F5> would give access to the list of files: Dir C:\WP51*. (see image).

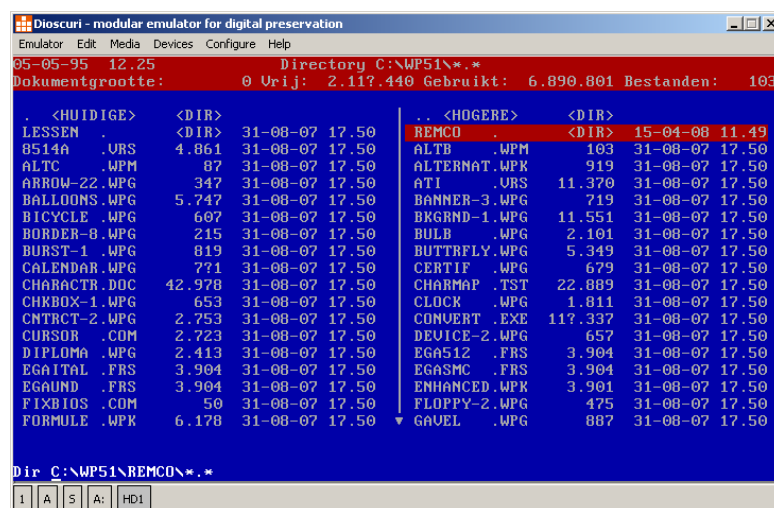


Figure 8: Screenshot WordPect 5.1; open file

By selecting the right directory (HUIDIGE [=CURRENT] or HOGERE [=HIGHER]) it is possible to navigate to the file to be opened <insert image>. It is necessary to use the cursor keys on the keyboard and the <Enter> key to confirm. The directory 'Remco' holds the WordPerfect files to be rendered. The selected WordPerfect file is opened by choosing option 1 (Opvragen [=Open File]).



Figure 9: Screenshot WordPect 5.1, rendering digital object

Next step is to establish the accuracy of the rendition of the WordPerfect file using Dioscuri compared to the 'original' rendition. Being in the luxurious position of having an obsolete laptop, a Compaq Contura 430C, running Windows 3.1 and WordPerfect 5.1, it was assumed that a comparison could be made between the emulated version of the WordPerfect file and the 'original rendering' on this obsolete hardware.

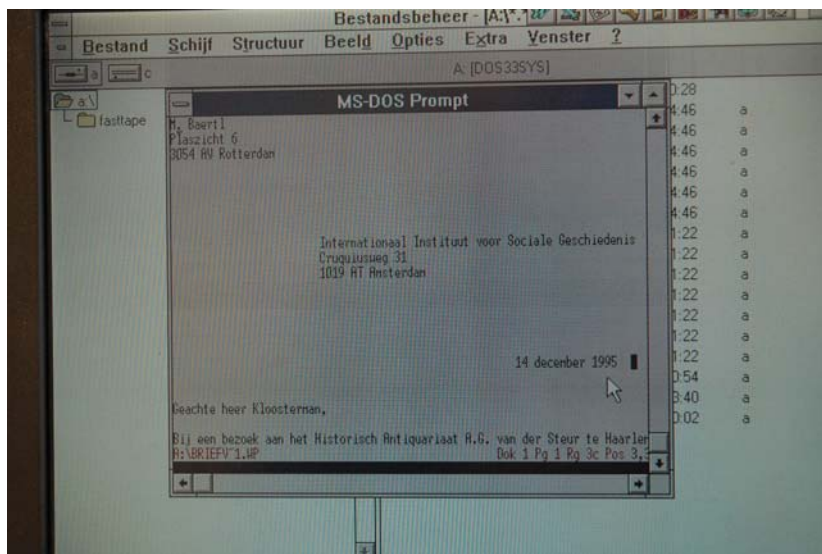


Figure 10: WordPerfect file rendered on Compaq Contura 430C



Figure 11: Compaq Contura 430C, running Windows 3.1 and WordPerfect 5.1

[3.2.6 Evaluate Experiment](#)

[3.2.6.1 Downloading, installing and configuring Dioscuri](#)

Downloading, installing and configuring the chosen emulation tool (Dioscuri, version 0.3.0) went quite easy following the manual on the Sourceforge website. Creating the required image was more difficult as we did not succeed in adding the WordPerfect data files to an existing image using Windows XP. We did succeed in adding these files using Linux. This was due to a lack of knowledge how to use WinImage under Windows XP.

[3.2.6.2 Operate WordPerfect](#)

It should be noted that it is anything but easy for future users to use old software. Future software will probably have a different appearance onscreen and will require a different approach to its use. This is demonstrated by the manner in which applications worked – and documents were prepared – before the emergence of the Graphical User Interface. One example is WordPerfect 5.1, which was very popular in the latter half of the 1980's. This application required the use of a wide variety of key combinations to create and use documents. There were more than forty combinations, and for this reason a card template for the keyboard indicating the combinations was supplied with the software. We experienced difficulty working with this old program, just 20 years after it was in daily use – even those who at the time were thoroughly familiar with the application. It is important to preserve the user manual of these old applications as well.

[3.2.6.3 Authenticity](#)

Strangely enough the representation of the WordPerfect files using Dioscuri is more authentic than the rendering of the WordPerfect file using the obsolete hardware. Under Dioscuri the WordPerfect background colour is blue (we all remember this famous WordPerfect colour), while using the laptop the background colour is grey. Under emulation the font colour is white, using the original hardware the font colour is black! We think that although running under MSDOS, the overall operating system on the laptop, being Windows 3.1, is transforming the colours as well as the shape of the characters. The laptop simply was too modern to render the files authentically. A real 16 bit 8086 pc running MS DOS as operating system is needed in order to make a true comparison between the emulated and the original environment.

[3.2.6.4 Full screen](#)

It was not possible to switch to full screen mode using Dioscuri, nor using the old laptop. The knowledge what combination of key strokes to use to toggle between full screen mode and reduced screen mode had evaporated.

[3.2.6.5 Compare emulated version with version within original environment](#)

The laptop that was used as reference appeared not to be fit for doing the job. This Compaq Contura 430C simply was too modern to be able to render WordPerfect 4.2 and 5.1 documents

authentically. This Contura has an Intel 80486 CPU, and was running Windows 3.1. And although Windows 3.1 can run MS-DOS programs, it is suspected that Windows 3.1 is influencing the rendition of WordPerfect files.

3.2.6.6 [Develop, test emulator and create images now](#)

Given the problem of finding a good way of comparing the emulated environment with the original environment, it is recommended to develop and test the emulator and to create images of the required operating system and application software (including digital objects) while the original hardware and software environment is still available, making sure that the rendering of digital objects using emulation can be compared against the 'original' rendition.

3.3 Case study 3: Legacy Map Software

3.3.1 Background

The BL's Map department receives many items that combine "data" (which may be images, geospatial databases, maps, etc.) with "programs" (to provide a front-end to that data). The software programs are for many of the items an essential part of the experience; for example, the data may be stored in proprietary formats, or the application might provide ways of locating and using the images. Typically these are Windows programs, but many of them are quite old and suffer incompatibility issues with recent versions of Windows. As such, there is a desire to provide suitable emulation environments to allow these programs to be installed and their data used.

Two specific programs were examined for the purposes of the case study, the Atlas Schoemaker and Panairama. The Atlas Schoemaker contains digitized images of the 18th century historical-topological atlas assembled by Andries Schoemaker, and a Windows front-end to search and view those images. The front-end requires a software component for manipulating images that shipped with some versions of Windows; Windows 95 OSR 2 (1996), Windows NT 4 (1996), Windows 98 (1998), Windows 2000 (2000), and Windows Me (2000). This software component was licensed by Microsoft from Wang, later bought by Kodak (and then various others). As of Windows XP (2001), MS stopped including it with Windows. Consequently, no currently shipping version of Windows can run the software. Library IT policy requires the use of Windows XP; as such, the reading rooms are no longer able to offer the Atlas Schoemaker to readers who would like to use it.

Panairama contains digitized aerial photographs of the city of Perth, Australia, along with a front-end for viewing the images. The versions of Windows supported are Windows 3.1 (1992), Windows 3.11 (1993), Windows 95 (1995), Windows 98 (1998), and Windows NT 4 (1996). Windows 2000, Windows XP, and Windows ME are all explicitly listed as not being supported. As with the Atlas Schoemaker, the software is currently unusable in the reading rooms due to the Library's IT policy.

3.3.2 Define Basic Experiment Properties

The goal of the experiment was to establish whether current commodity virtualization software would allow the use of legacy PC operating systems to enable access to the maps software described above and, if suitable, how much expertise was necessary to successfully set up and install the virtual machines.

Although it would be clearly desirable to compare the virtual machines with "real" hardware running the software, this was unfortunately not possible due to a lack of such hardware; instead, best guess appraisals were made according to our own experience and expectations.

3.3.3 Design Experiment

The strategy for each program was to install the virtualization software, to install a suitably old operating system within the virtual machine, and then to install the programs themselves. Due to previous experience, the virtualization software used was VirtualBox⁹ and Microsoft Virtual PC 2007¹⁰. It was decided to use Windows 98SE for Atlas Schoemaker and Windows 3.1 for Panairama. Although in principle both could have used Windows 98, it was felt that using different operating systems would give more useful feedback.

The success criterion for the experiment was whether it would enable the use of the software on a modern PC.

3.3.4 Specify Resources

Both experiments were performed on a recent desktop PC; Windows Vista Ultimate, Core 2 Duo E6600, 4 GB RAM.

3.3.4.1 Atlas Schoemaker

The chosen target operating system was Windows 98SE; any of the compatible operating systems would suffice, but Windows 98SE was more readily attainable than the others. Due to distribution of the operating system being contractually prohibited, it had to be specially obtained from Microsoft, in the form of a Windows 98 Virtual PC image. As such, the decision on which virtualization

⁹ <http://www.virtualbox.org/>

¹⁰ <http://www.microsoft.com/windows/products/winfamily/virtualpc/default.mspx>

platform to use was made 'automatically', and the image was pre-created (requiring no effort to install the software). Installing Virtual PC took only a few minutes, and was very simple.

The image as delivered was functional and sufficient for the testing. Some minor configuration adjustments were made (screen resolution, colour depth) and then the software was installed. The software installed cleanly without any problems. Windows 98 is sufficiently similar to current versions of Windows that it should be sufficiently familiar to perform these tasks without significant prior experience.

[3.3.4.2 Panairama](#)

It was decided to test the software on Windows for Workgroups 3.11 on MS DOS 6.22. Windows 3.11 is readily available from Microsoft (it is not subject to any contractual/legal restrictions) and it was felt that this combination would be a good representative of some of the older operating systems.

Two virtualization products were used for the testing; VirtualBox and Virtual PC. Both of these claim to offer some amount of Windows 3.11 support.

[3.3.4.3 VirtualBox](#)

Installation was in both cases slightly troublesome; Microsoft distributes copies of the DOS 6.22 *upgrade*, but the upgrade will not install, unless a version of MS-DOS is already installed. Fortunately, it is possible to trick the installer into thinking that MS-DOS is already installed by cancelling the installation, formatting and making bootable the hard disk, and then restarting the installation.

The next step was to install support for the (virtual) CD-ROM drive. There are several generic MS-DOS CD drivers available. The easiest to locate was the "oak" driver from a Windows 98 repair disk. This was widely used and is widely compatible with IDE CD-ROM drives, including the virtual CD-ROM drive presented by the virtualization software. Configuration information for the driver was copied from the start-up files of the Windows 98 repair disk.

Installing Windows took a little effort. Panairama requires 24-bit colour to display images properly (they are full-colour photographs) and recommends SVGA resolutions (i.e. greater than 640x480), but the video drivers supplied with Windows 3.11 were not able to use VirtualBox in such a way. It is possible to modify the drivers slightly to allow VirtualBox to support 16-bit colour, but this was later found to be insufficient to properly view the images.

[3.3.4.4 Virtual PC](#)

Virtual PC 2007 was more satisfactory. The initial set-up process (for DOS and the CD drivers) was identical to VirtualBox; the difference laid in the Windows drivers. Although Virtual PC 2007 no longer includes Windows 3.x drivers, an earlier version (2004) did, and this earlier version can be freely obtained from MS. The Windows 3.x driver package from Virtual PC 2004 can then be used with virtual PC 2007. The drivers allow high-resolution (1024x768) full-colour (24-bit) graphics within Virtual PC.

Installing the software itself was unproblematic. Although Windows 3.x is very different from current generation operating systems, the instructions supplied along with the software were sufficient to get things going.

The time taken to actually create the images was not long in itself; no more than a couple of hours for each virtualization platform. What did take the time was finding the drivers (especially for VirtualBox; although unnecessary for this task, the Microsoft TCP/IP stack for Windows 3.11 was installed, which required some hunting to find the right network drivers and configure them properly) and finding the right settings to configure them properly, along with knowing the tricks to get MS-DOS to install. Virtual PC was significantly easier in this regard, because all the drivers that are necessary can be found in the Virtual PC 2004 package. It took some days of reading and experimentation to get a satisfactory VirtualBox image (and even then, its video drivers are not really usable for this application).

[3.3.5 Run Experiment](#)

The general procedure when using the programs was to attempt to ascertain whether they worked properly. It was attempted to install both of them in Windows XP to see if it was possible to establish some basis for comparison. In the case of Panairama, installing the software on Windows XP worked, but in usage the software failed to load, citing an inability to load a "16-bit VBX control",

and then crashing. Atlas Schoemaker installed on XP but produced errors during use and was unable to display any images. In both cases, therefore, the emulated software experience was significantly superior to the attempt to use the software on an unsupported operating system.

3.3.6 Evaluate Experiment

3.3.6.1 [Atlas Schoemaker](#)

In usage, the software seems to run properly. There was no "real" Windows 98SE machine to compare with (so cannot be 100% certain that it is working as it should), but the key functionality (searching for and viewing images) worked as expected. Performance was good; loading each image took a second or so, and searches completed in a few seconds. Again, it is hard to know how this compares with original hardware, but it was certainly no impediment to using the application. Accordingly, the testing performed was very informal; the features of the application were used to see if they worked. Unfortunately, it was not possible to perform e.g. benchmarks against the real thing.

3.3.6.2 [Panairama](#)

As with the Atlas Schoemaker, everything appears to work normally with good performance, but there were no real means of verifying. As with the other software, the testing was necessarily informal, having nothing to compare against.

3.3.6.3 [VirtualBox and Virtual PC](#)

In both cases, the results of the virtualization seemed satisfactory; the performance was good, and the software ran as it should have. As such, the experiment was certainly a success. A recurring problem was a lack of baseline to compare to—the experimenter had not used either program on "real" hardware, so if there were significant discrepancies, it was not possible to establish. It is suspected that there were not, as both programs appeared correct, but one cannot be sure. One slight concern is that the support for Windows 3.x is dying off even within the (commercial) virtualization products. Virtual PC 2007 no longer ships with the essential driver package that Virtual PC 2004 contained, and although the drivers from the latter are compatible with the former, it took some digging to discover this. It would be quite possible for future versions to drop support for MS-DOS and Windows 3.x entirely.

3.3.6.4 [General comments](#)

One thing that was noticed was how dependent the experimenter was on the knowledge he had acquired 10-15 years ago, when using Windows 3.x and (to a lesser extent) Windows 98 "for real". Although rusty, the experimenter knew how to configure them, he knew how to trick MS-DOS into installing, he knew where to find an IDE CD-ROM driver--because he had done it all before. Without that knowledge, it would have been extremely hard to successfully install the operating systems and the software. Once the operating systems were in place, installing the software was the easy bit. There were some problems with the virtualization itself (the lack of suitable video drivers in VirtualBox, for example), but for the most part the virtualization platforms did what they were supposed to do with the minimum of fuss.

Preserving the techniques and tricks used to get Windows 3.x/DOS working seems to be more of a challenge than getting the software to run. It is difficult to see how Planets might begin to address this; it is a human problem, not a technology problem.

3.4 Case study 4: Execution of Different Computer Games

3.4.1 Background

Computer games have been around for many years. They emerged on the first commercially made machines produced in the mid end of the 1960s. The first great wave of game titles arrived with the introduction of the first so-called “home computers”¹¹ and arcade games at the end of the 1970s. Beginning of the 1990s several thousand game titles for the different computer and games systems were available.

With the dominance of the pc in the early 1990s, the gaming scene moved mostly over to the x86 architecture. The computer games were programmed for the most popular operating systems on this architecture: MS-DOS, DOS-based Windows and later the Windows-NT based product line. Mid/end of the 1990s a new era of game consoles emerged. These were handheld devices or specific machinery to be connected to the TV set.

Computer games are part of our digital heritage, and so in many countries national libraries have collected them. A strategy is therefore needed to preserve these objects in such a way as to give access to them.

Computer games make good tests of the characteristics of emulation software because they often make extensive use of the computer’s hardware (especially audio and video output and user input), as well as being performance-sensitive. For the first round of tests some popular games of the 1980s and 1990s were selected from the software archive¹²:

- “Arkanoid”. Games like this represent the early home computer era. Game produces graphical output and sound.
- “Loom” and “Indiana Jones and the Fate of Atlantis”. LucasArts produced many similar adventure games using the same underlying technology platform, offering a mix of sound and graphics.
- “DOOM” was a landmark game. It was one of the first first-person shooter games and was the first in a successful series of games. As well as 3D graphics, it also supports network play.
- “Myst” is an adventure game similar to Indiana Jones and the Fate of Atlantis. Until 2002 it was the best-selling PC computer game of all times, renowned for its involving puzzles and attractive graphics.
- “Command & Conquer” is a real-time strategy game that runs in both MS-DOS and Windows. It offers network play.

The list of selected games should be extended for further experiments.

3.4.2 Define Basic Experiment Properties

What is needed to actually run an ancient computer game? For what kind of platform the game was written (the number of game platforms is rather huge compared to the typical PC era applications)? What kind of environment is needed? What additional software components are required?

Is the rendering complete (graphics and video output in decent quality)? What is the viewing experience (full screen possible, decent speed)? What about interaction (proper input method mapping)?

What kind of documentation is needed? Handbook of the game itself? Documentation of the platform/operating system? Documentation of the emulator involved?

Furthermore: What are the strengths and weaknesses of emulation in general and the chosen emulation tool in particular (in this case MESS¹³, VMware¹⁴ and QEMU¹⁵)? What are the lessons learned? What aspects of emulation or emulation tool require further research in the next iteration?

¹¹ <http://www.planets-project.eu/private/pages/wiki/index.php/HomeComputer>

¹² <http://www.planets-project.eu/private/pages/wiki/index.php/EmulationSoftwareArchive>

¹³ <http://www.mess.org/>

¹⁴ <http://www.vmware.com/>

¹⁵ <http://fabrice.bellard.free.fr/qemu/>

3.4.3 Design Experiment

The set of emulation options is different for the sample of games. The Commodore 64 game, Arkanoid, requires an emulated variant of the specific hardware the game was written for. The selected emulator, MESS, is able to emulate a wide selection of machines, among others the Commodore 64 and its variants.

The requirements of the pc games are twofold: They might require a certain set of hardware features (CPU speed, screen resolution and colour depth) but depend on certain API of the operating system too.

Thus the strategy for each game was to install emulation or virtualization software as the base. Next a suitable older operating system, in this case Windows 95 and then Windows 98, was installed. The differences between both operating systems were marginal, thus the latter one was selected.

For the Commodore 64 a two-step approach was taken: For testing purposes, the MESS was installed to a Windows 98 executed within a VMware image (emulation stacking). This approach makes it possible to exchange a pre-prepared image easily with other institutions. Additional information on how to start a game, how to use the emulator and digitized PDFs of manuals/handbooks could be added. The same would be possible with Windows XP or Linux, but with a somewhat larger disk footprint.

The success criterion for this experiment was whether it would be possible to actually play the old games within a modern PC environment (Windows and Linux).

3.4.4 Specify Resources

3.4.4.1 Preparation

The main tests were performed on two different system environments; Linux (SuSE 10.2, later repeated on Ubuntu 8.04) and Windows XP Professional, using standard hardware such as an AMD CPU desktop system with 1 GB of memory, Dell 860 series laptop with dual-core CPU and 1.5 GB of RAM running Windows XP SP2, and a Lenovo X60s laptop with dual-core CPU and 1.5 GB of RAM running SuSE Linux Version 10.2 (Kernel 2.6.18). It took about half a day to setup the operating systems, VMware and QEMU.

The construction of images and installing the guest operating system took another half day. The images are special files mimicking a virtual block device for the emulators. It would be possible to use real hard disk partitions for VMware, but that is of no much use within the context of long-term preservation. VMware offers virtual SCSI and IDE drives. IDE is often the better solution due to wider compatibility with old operating systems.

Container disk images are convertible—the qemu-img program can convert several image types into each other, provided that IDE images are used.

The conversion experiments (goal: to convert containers into other formats) took about half a day, because of testing them afterwards. If you change the virtual machine in most cases you will have to change the drivers of the installed operating system. All significant drivers are stored in the named software archive.

All digital objects were already taken from their original media and copied to the software archive.

3.4.4.2 Assess emulation options

There are several emulators available for the Commodore 64 machine, e.g. MESS and JaC64¹⁶. MESS was selected because of its greater popularity and support for hundreds of other home computer architectures and variants.

3.4.4.3 Select emulation solution for testing

It was decided to perform emulation of hardware, using x86 with QEMU, VMware and DOSBox.

As a test for comparison, ScummVM¹⁷ was used for the LucasArts games. ScummVM is special emulation software which is able to interpret the LucasArts games file formats. The scummVM is an application emulator using a higher level emulation approach compared to DOSBox (operating system/hardware) and QEMU/VMware (hardware). It is possible to start exactly the same game

¹⁶ <http://www.jac64.com/>

¹⁷ <http://www.scummvm.org/>

within two different emulators and compare the visual and sound output and the keyboard/mouse input. At the same time full screen modes and execution speeds could be evaluated and compared. In this setup ScummVM could be used as a proxy for the original PC environment (which was not present or too laborious to setup).

3.4.5 Run Experiment



Figure 12: “Myst” of Broderbund Software running within VMplayer 2.0 running Windows 98

The experiment started "from scratch" setting up the selected emulator/virtualization tool and installing the required operating system. Tests were performed with Windows 95 and Windows 98. Windows 98 was selected because there were no significant differences. The installation of the PC games was part of the experiment. The most typical setup would be to put the medium (a copy of the original medium to a file using a virtual drive) into its drive and start the emulation.

The configuration of the virtual machines in VMware is rather straightforward. With QEMU the configuration of the virtual machine is specified via the command line¹⁸.

The installation of the operating systems worked well with the stored floppy images within the software archive. Booting from CD for installation was not successful in every case. The setup itself was a rather fast procedure, but it is necessary to have the license numbers ready to be able to complete the procedure.

The installation of the game was straightforward from the CD images or floppy disks. Any additional software that was needed, such as DirectX for Command and Conquer: Red Alert, was typically shipped with the game itself.

It was tried to use an older laptop as a hardware reference platform. But it appeared to be too difficult to understand the procedure to change the BIOS boot order to get the operating system installed to it (Toshiba Tecra, PII, 64 MB of RAM).

Additional tests were done with the Fate of Atlantis in ScummVM and DOSBox and with DOOM in DOSBox.

3.4.6 Evaluate Experiment

The overall experiment results were very promising. It was possible to recreate the needed environments to run the old games within modern environments. The overall experience of interaction was successfully replicated with some minor modifications.

For all games it was possible to achieve a full screen experience: Typically the game screen output covered the whole computer screen. Both the tested virtualization and emulation tools for x86 hardware and the emulator MESS for the Commodore 64 (and other systems) are able to run in full screen mode.

A bigger issue is the input mapping. In order to get proper translation of input, the keyboard layout in MESS has to be switched. Otherwise the "*" needed to load the game is not properly translated and thus not able to be typed.

¹⁸ http://www.planets-project.eu/private/pages/wiki/index.php/Qemu_scenarios

Documentation on usage is needed. How to configure and use the emulator? There are several related projects (like the emu loader) which provide additional meta information (pictures of hardware, peripherals, boxes and packages, artwork, screenshots, etc.). How to install and load a certain game? How to start and run it? Not all buttons for steering an application are named during start-up. The key mappings might differ from the experience of today's keyboards. Many procedures are command line orientated.

There are different formats of game images. MESS is able to cope with floppy, cartridge and tape images. The user should know how to handle and start each. For instance, cartridges start automatically, games from a floppy image are to be loaded and run manually.

ROM images for the target systems are needed. That might be a problem because the original hardware should be present to have the rights to use images. For instance, the computer games museum in Berlin has a large collection of different systems, so virtually the needed items were borrowed. Many memory institutions most probably have this kind of machinery somewhere in stock.

In general, the results of emulation were satisfactory to a large extent. The operating systems and the installed games behaved as expected. Both persons who run the experiments had greater knowledge of the used operating systems, but the knowledge on the Commodore 64 (home computer) and its games was rather buried and had to be extracted via extensive web searches. Some of the needed information was just copied into files of the environment (at the same level as the MESS emulator was installed), so any average user should be able to reproduce the results. It was not possible to run the experiments on original hardware. That would have been too much effort and it was obvious that the results of running Windows 95/98 were very well comparable, as hardware input and output is not too different at the moment. Problems were expected with the availability of drivers for the operating system over time. For instance, VMware completely dropped driver support for Windows 95/98 starting its 4.0 line of Workstation. In that context QEMU might be the best solution for x86 as a wide range of real hardware is supported. Even ISA only machines and more modern hardware setups could be configured.

One major problem is the implicit knowledge (see comments on the other case studies too). The experimenters were familiar with the things to do. In earlier experiments quite some problems were experienced when installing and configuring Apple operating systems, due to a lack of experience with this specific operating system. Therefore, it might be of interest for future experiments or case studies to have a younger computer expert performing these experiments again. At this moment a large pile of paper (old handbooks of some of the software deposited in the archive) is stored, but there was no need to really use it. Nevertheless it will be useful to have this information online. Unfortunately not all the user guides or manuals for the games were available. As a result, it was not possible to get past the "license" check of the "Fate of Atlantis" game using VMware/QEMU/DOSBox. In the interpreter mode of ScummVM this test was skipped. The game "Loom" was deployed as a second test, because a manual of "Loom" was available. The test revealed that the keyboard input mapping of DOSBox was correct (proper translation of German key mappings to actions), while the execution speed was quite satisfactory in both settings.

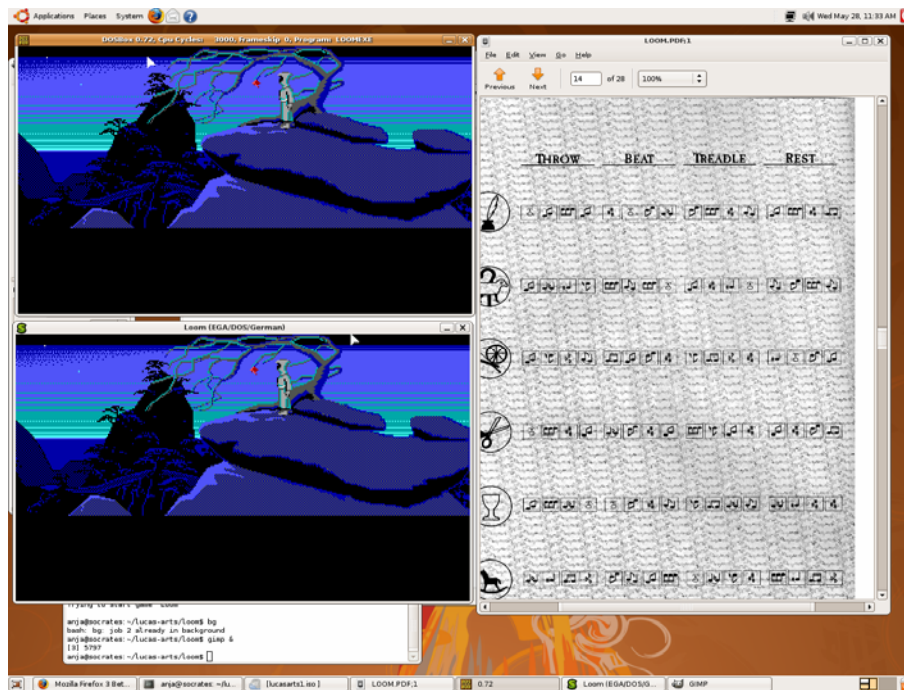


Figure 13: "Loom" running in two different emulators: DOSBOX and ScummVM.

3.5 Case study 5: Preserving interactivity with digital objects

3.5.1 Define Basic Experiment Properties

The Universal Virtual Computer is a specification of a simple computer with a limited instruction set. The assumption is made that a new emulator for this computer can be developed on future computers. Emulators for the UVC have been developed in both Java and C++. It takes about 1 to 2 man months to create the emulator. Previous work showed how format interpreters written for the UVC are used to convert binary objects (like GIF, JPEG) to a human readable/understandable format (text). As the format is understandable, now and in the future a restore program can be developed that recreates the presentation of the digital object.

This previous research handles only static features of the preserved objects and does not involve general interaction with the outside world. In this case study, interaction with a preserved digital object is developed and tested.

3.5.2 Design Experiment

The first decision to make was what kind of application to preserve. A whole spectrum of applications exists, from a simple text editor, like notepad, to complex CAD applications (in both the GUI and the mathematical models) and highly interactive games. Neither end of this spectrum is a good start for this use case study. This case study is focussing on a spreadsheet for two main reasons: first spreadsheet applications are well known and widely used, secondly a spreadsheet contains both data and a mathematical model.

Key in the design of the UVC program is to formalize all interaction so that it no longer depends on a particular I/O device. The UVC program should not depend on a disk, mouse and/or keyboard.

3.5.3 Specify Resources

The first step for preserving the interactivity with digital objects was selecting the specific application to preserve. The early ambition was to work with Open Office Calc. Our conclusion was that the application is not solely written in Java, it actually consists of a collection of tools. The design and the Java implementation of the Sharp Tools Spreadsheet suited better for preservation with the UVC. Having the Java source available is important as the only higher programming language supported for UVC program development is a Java subset. This subset covers the mini Java specification with a number of additions, like the support for Floats.

In previous research a C++ version of the UVC emulator was developed. This C++ version is used in this case study. The main portions of the development were the building of the UVC program by extracting the logic from the Sharp Tool Spreadsheet application and the construction of the restore program.

3.5.4 Run Experiment

The design and development work was performed by one developer with the supervision and support of the other IBM Planets members. The total project took about 5.5 months. The following diagram shows the component model of the solution.

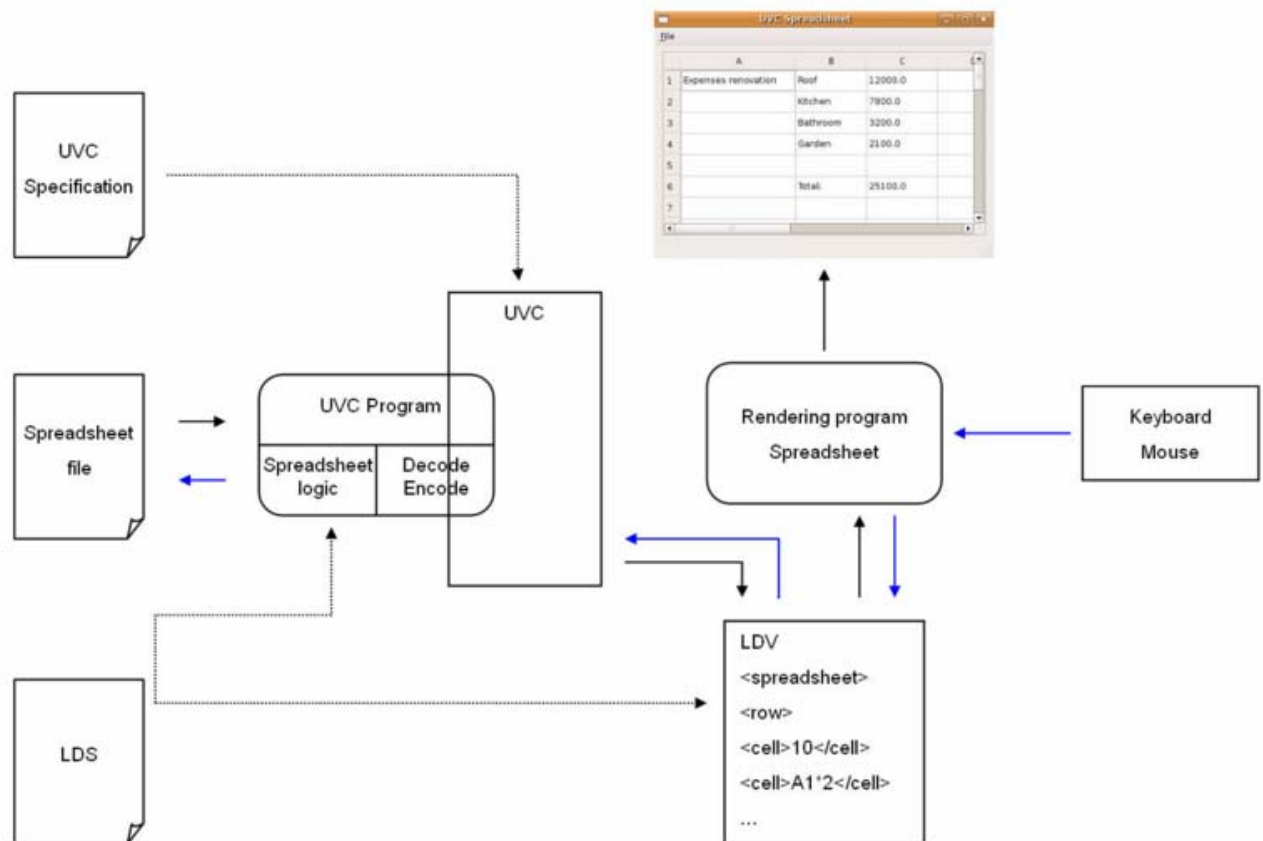


Figure 14: Component model of UVC

The UVC Specification is a document that specifies the UVC. Based on the specification an UVC emulator is created. This UVC emulator is capable of executing an UVC program. In this case study the UVC program contains the logic for decode/encode the original file format and the recalculation of the spreadsheet. The Logical Data Scheme defines the layout of the messages from the UVC program to the outside world. The Logical Data View (LDV) is an actual message conform the LDS. The restore program connects with the UVC and contains the platform specific implementation details for display and input from mouse and keyboard.

3.5.5 Observations

We start with the key point in our case study, the communication channel. The UVC program and the restore program must interact through messages independent of the underlying protocol.

The restore program and the UVC program are using the messages as defined in the following table.

Front-end to module

type	description	data
1	output dimensions	-
2	output grid	-
3	select cell	int[]
4	clear cell	-
5	set input value	char[]
6	get value	-
7	get input value	-
8	output file	-

Module to Front-end

type	description	data
0	file loaded	-
1	dimensions output	int[]
2	location output	int[]
3	select confirmation	boolean
4	clear confirmation	boolean
5	value confirmation	boolean
6	value output	char[]
7	input value output	char[]
8	file output	char[]

Figure 15: Messages used by restore and UVC program

The design of the messages went fine. The first observation is that the specification of the communication protocol meets the requirements as defined for this case study.

The goal of this case study was to preserve interaction with a digital object. A restore program is developed in C++ using the Qt libraries on Linux. The Sharp Tools Spreadsheet is developed in Java and using the Swing libraries. The second observation is that the restore program looks like the Sharp Tools Spreadsheet, but the details of the look and the interaction do differ due to the fact they are developed using different toolkits.

The third observation is that the level of interactivity of the original program and the required interaction in the restore program determines the amount of effort required to develop the UVC program and the restore program. The Sharp Tools Spreadsheet provides less interactive functions than Microsoft Excel. For example, in Excel it is possible to fill two adjacent cells with values that are the beginning of a range (like 1,2), then select the two cells and use the mouse to extend the box around the two cells, the new cells in the box will be filled with 3,4,5, ... Redeveloping this kind of presentation layer functionality in the restore program requires significant effort. The UVC approach stays viable as long as the focus is on the business logic of the application and not the presentation itself. The restore program for the Sharp Tools Spreadsheet might be a good restore program for an Excel spreadsheet as well.

The last observation is about the tools and compilers available to develop UVC programs. Currently a compiler is used that compiles code that complies to the mini-Java specification with a number of additions. For the compilation of the code (the extracted business logic) from, for example the Open Office suite, more tools are required.

3.5.6 Evaluate Experiment

The Sharp Tools Spreadsheet case study has shown the feasibility to preserve application logic and to recreate interaction with preserved digital objects. The specification of the communication channel provides good support for developing interaction between the UVC program and the restore program. The strong point is that the UVC program has no dependencies on currently used devices.

A success factor for this preservation approach is the tooling available to create UVC programs. Currently the Java compiler to UVC compiler has been developed. To support a wider range of applications the planning is to connect to a portable compiler project, like the Amsterdam Compiler Kit. Having a UVC compiler that compiles intermediate code from such project into UCV assembly will make a wider range of programming languages available to the UVC.

4. Conclusions and lessons learned

4.1 General

The performed case studies as presented in the previous sections show a wide variety of applications for emulation in the field of long-term preservation. Each case study addressed a particular situation in which emulation or virtualisation can support access to a digital object in its original computer environment. Although the number of case studies presented in this report does not cover all possible cases in which emulation or virtualisation can be useful, the cases were chosen as such to give a broad view on the different objects that are preserved by memory institutions.

For each case in this report, emulation and/or virtualisation have proven to be a viable preservation approach that is worthwhile investigating. In all case studies the results of applying emulation and virtualisation tools were promising. The experiments showed that emulation and virtualisation tools were capable of rendering WordPerfect 5.1 documents, websites from the early internet era, legacy map software, old computer games, and extraction of valuable information from spreadsheets. Of course, problems and disadvantages were found as well. Therefore, a next iteration of experiments with case studies would be useful that cover new cases and a deeper investigation of existing cases.

4.2 Lessons learned

In general, the following lessons have been learned:

1. **Authenticity.** To ensure that emulation of an original object is accurate, the emulated environment should be compared with its original environment. However, in some cases this is not possible anymore as the original hardware no longer exists or no longer works. Therefore, comparison experiments between emulated and actual environment should be performed, while the actual environment is still active. This also has been shown by the UVC case study. The UVC can not guarantee that it mimics the exact interactivity and therefore has to be tested at the moment the target environment is still functional.
2. **Emulator preparation.** Although emulators and virtualisation tools have existed for many years, these tools still require special skills to operate. The emulator often has to be configured manually and software that should run on the emulator has to be prepared as well. Disk image manipulation tools are needed to create and setup the targeted operating system, additional software, plug-ins, fonts and the digital object(s). This preparation is difficult and requires a better (automated) approach than currently available.
3. **Media migration.** Almost every emulator and virtualisation tool seems to rely on virtual media. This means that the actual bits must be transferred from physical carrier onto a virtual disk image before they can be accessed via emulation. But a lot of content in libraries and archives still reside on the original data carrier.
4. **Software preservation.** Currently, the focus is primarily on preserving digital objects residing at libraries, archives, data centres and museums. However, applying emulation requires recreation of the whole computer environment. While the emulator mimics the hardware virtually, software such as operating system and user applications, drivers, plug-ins, font sets, etc. is needed to render the digital object. This obviously requires preservation of software. Some libraries and archives are already doing this, but a structural and coordinated approach is missing.
5. **Legal implications.** Technically, emulation proposes a good way of retaining access to digital objects of which the hardware is no longer available. From a legal perspective, emulation might violate software copy protection mechanisms as data may need to be copied from physical carrier to a disk image. Furthermore, software licenses may be needed to use certain applications under emulation. This might be more difficult than expected as some vendors may have gone out of business while they still hold the rights on their old products.

6. **Computer skills.** Operating a computer requires knowledge of hardware and software. For current computers this seems quite trivial, but using an old computer such as an MS-DOS machine running WordPerfect 5.1 is different. Special knowledge is required which is often only available in people's mind or in tutorial books and manuals. Preservation of this knowledge is important, and active efforts must be made to preserve it whenever new hardware and software emerge.

4.3 Further research

Based on the conclusions and lessons learned, several research topics are identified that require further investigation within the Planets project and beyond.

4.3.1 Disk image preparation

Many emulation and virtualisation tools only support virtual media, which means that disk images have to be used containing all required software such as OS, user applications, plug-ins, drivers, font sets and more. The difficulty is that each digital object might require a different setup of the disk image. To prevent a user from having to manually setup the computer environment each time an object should be rendered, several methods for compiling these images could be considered:

- store a disk image containing pre-installed application software and operation system. This is the fastest way for emulators to create a target environment, but requires a very large amount of storage space as each digital object requires a unique disk image
- store a simple set of images containing only an operating system 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 operating system, along with a differential bit stream containing information of application software installed onto that image (this can easily be created by 'subtracting' an operating system-only image from the operating system plus application image). When required the environment can be recreated by merging the differential bit stream with the original operating system image.

Currently, only the first method or a combination of the second method is used, but none of them are automated or efficient. A deeper investigation into this topic is desirable.

4.3.2 Software preservation

As software is one of the most important aspects for recreating an old computer environment, coordinated research is needed into the technical, organisational and legal issues around preservation of software for emulation. It is crucial for the widespread application of emulation to create a (central) software archive of images with different (obsolete) versions of operation systems and (obsolete) application software, in different languages that the digital preservation community can make use of. This also means we have to deal with the issue of software licences.

4.3.3 Remote emulation

All case studies described in this report were setup locally and required extensive preparation before the original digital object could be experienced via emulation or virtualisation. To reduce complexity and improve user friendliness and interoperability within the Planets framework, emulation processes should be run remotely in such a way that the user can experience the rendering process locally, while the configuration and execution of the emulation process takes place elsewhere. This topic is one of the major research activities within PA/5.