



Project Number	IST-2006-033789
Project Title	PLANETS
Title of Deliverable	Framework and Workflow for testing tools for different technical environments
Deliverable Number	PA/5-D1
Contributing Sub-project and Work-package	PA/5
Deliverable	Internal
Dissemination Level	Public
Deliverable Nature	Report
Date	30 June 2007
Author(s)	BL

---

## Document Control

### Contributors

Person	Role	Partner	Contribution
Paul Wheatley	PA/5	BL	Document owner and co-author
Jeffrey Van der Hoeven & Remco Verdegem	PA/5	KB-NL & NANETH	Co-authors
Dirk von Suchodoletz and Raymond van Diessen	PA/5	ALUF & IBM	Comments and review

---

## Table of Contents

1	<i>Aim</i> .....	4
2	<i>About emulation</i> .....	4
2.1	What is emulation?.....	4
2.2	Pros and cons of emulation .....	4
2.3	Emulation versus virtualisation .....	5
2.4	Emulation implementation.....	5
2.5	Level of emulation .....	6
3	<i>Previous experiences in emulation testing</i> .....	6
3.1	Enthusiast experience.....	6
3.2	User Testing.....	7
3.3	Visual testing/comparison of specific criteria .....	7
4	<i>Approach taken by Planets for emulation testing</i> .....	7
5	<i>Emulation uses</i> .....	8
6	<i>Purpose of emulation testing</i> .....	10
7	<i>The Planets approach to emulation testing</i> .....	10
7.1	Workflow for emulation experiments.....	10
7.2	The role of the Testbed in the execution of emulation experiments .....	12
7.3	Research Questions.....	12
7.3.1	Authenticity .....	13
7.3.2	Strengths and weaknesses of emulation.....	13
7.3.3	Effectiveness/appropriateness of a tool.....	13
7.3.4	Metadata .....	13
7.3.5	Technical issues .....	14
7.4	Costs .....	14
7.5	Experiment Strategies.....	14
7.5.1	Visual Comparison.....	14
7.5.2	Automated Comparison .....	14
7.5.3	User Testing.....	15
7.5.4	Benchmarking.....	15
7.5.5	Execution Testing .....	16
7.6	Experiment Criteria .....	16
8	<i>Risks and dependencies</i> .....	19
8.1	Technical preparation issues .....	19
8.2	Timeliness .....	19
9	<i>Emulation experiment example: Educational multimedia application</i> .....	20
10	<i>Conclusions</i> .....	21

---

## 1 Aim

The purpose of this document is to describe a strategy, framework and workflow for testing emulation within the Planets Project. The focus of this work is on evaluating emulators and their effectiveness or appropriateness for preserving digital objects. The framework will describe the practical strategies that will be used to perform specific emulation experiments, and how those experiments should be designed and executed according to a standard workflow.

The document is intended to describe an approach to emulation testing that will be useful to digital preservation experts considering the use of emulation as a preservation strategy, who need to evaluate the tools they are considering using. More specifically, it is expected to inform the next steps of the PA/5 work-package of the Planets Project, as it moves forward to practical experimentation and testing of emulators developed by Planets and emulators available from commercial companies and the open source community.

This document will consider the purposes of emulation and emulation testing, the strategies that can be pursued to implement testing, the strategies that can be used to evaluate the results of testing, and some discussion of the many technical, organisational and other issues which might need to be considered.

This document provides an introduction to the use of emulation in a preservation context, before examining previous approaches to emulation testing. Based on existing work, a Planets approach to emulation testing is then proposed. This includes description of strategies for the evaluation of emulation as well as consideration for technical issues. This approach is rounded off with an experiment example. Conclusions are then drawn and recommendations for the next phase of work are made.

This document will be revised as practical developments and experiences inform the initial approach to emulation testing proposed in this first version.

---

## 2 About emulation

---

### 2.1 What is emulation?

The term emulation is used in computer science to denote a range of techniques all of which involve using some device or program in place of a different one to achieve the same effect as using the original. The term "simulation" is often confused with--and sometimes even used as a synonym for--emulation, but we distinguish between the two terms here by noting that a simulation describes what some other thing would do or how it would act, whereas an emulation actually does what that thing would do. For example, an airplane simulator does not actually fly. That is, simulation generally involves the use of a model to understand, predict or design the behaviour of a system rather than the practical re-creation of that system's capabilities. In contrast, emulation is generally used to create a surrogate for the system being emulated. For preservation purposes, we focus on emulating older, maybe obsolete computers on current and future computers. In this context, emulation might enable future computers to mimic any obsolete computer, virtually re-creating the obsolete computer and thereby allowing its original, obsolete software to be run in the future. This would allow the original rendering programs for obsolete digital formats to be run on future computers, under emulation.

---

### 2.2 Pros and cons of emulation

Emulation has a number of key benefits, but also some disadvantages in development and use<sup>1</sup>.

---

<sup>1</sup> For an extensive description of the key benefits of emulation, see: statement Emulation Expert Meeting 2006, [http://www.kb.nl/hrd/dd/dd\\_projecten/projecten\\_emulatie-eem-en.html](http://www.kb.nl/hrd/dd/dd_projecten/projecten_emulatie-eem-en.html)

Advantages are:

- Allows access to digital objects in their original intended form, appearance and functionality.
- No periodic effort is required to keep the digital object up to date (thereby avoiding any risk of losing original data when it is changed or migrated).
- One emulator is a key to a wide range of digital objects in various formats and depending on various software environments.

Disadvantages are:

- An emulator has to be developed which can involve considerable effort.
- An emulator needs to be maintained over time as it too is in danger of becoming obsolete.
- Required software (OS, applications, plug-ins, etc.) have to be preserved<sup>2</sup>.
- Knowledge of the target environment have to be maintained (manuals, practices, undocumented experiences) .

---

### 2.3 Emulation versus virtualisation

Emulation is often confused with virtualisation. Although both methods allow the user to experience a different target environment running on a host machine, the underlying technique is different. To “virtualise” a certain computer environment, the virtualisation software uses a virtual machine to separate the host platform from the target platform. The software within the virtual machine is typically compatible with the underlying hardware. Thus the virtualisation 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 always 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 of a platform can provide a variety of benefits. Commercially, the most widespread reason for using virtualisation technology is to multiplex a physical machine; to provide multiple independent operating systems on a single physical instance. This has a number of advantages:

- isolation between different pieces of software
- ability to run multiple operating systems
- easier management of servers

Virtualisation is often faster than emulation, but is less flexible as it is very tight to the underlying computer system. Slight changes to the host system can lead to inaccessible target environments. Therefore, virtualisation seems to be a less durable approach to ensure access to digital objects for the long term. However, it is interesting to compare emulation with virtualisation and should be incorporated in the range of experiments.

---

### 2.4 Emulation implementation

Emulation can be implemented in hardware or software. Emulators in hardware are commonly used in embedded devices to offer backwards compatibility with previous devices. However, an emulator in hardware is not very flexible and will be difficult to maintain without an endless supply of spare parts. Most commonly used emulators are written as software programs that run on a computer and operating system.

---

<sup>2</sup> Software and digital objects are typically removed from the medium on which they were originally stored and captured as easily preserved bytestreams that capture the significant properties of the original. These “disk images” are used by emulators in place of the original disks or other media.

---

## 2.5 Level of emulation

Aside from their implementation, each emulator emulates a specific level of the emulated computer: hardware, operating system (OS) or application level. Most emulators emulate hardware. The reason choosing hardware emulation is that hardware is often very well specified as hardware components need to be manufactured. Therefore, precise specifications are required of which many of them are publicly available to ensure hardware compatibility. When emulating on hardware level any software program that is written for that platform can run on it, just like it does on real hardware. Emulating the operating system level allows the user to run applications written for an OS, but applications that call direct hardware routines will not run. OS documentation is often proprietary and this can hinder emulation at this level. Despite these drawbacks, emulation at a higher level than the hardware can offer simpler design and implementation where in effect the emulation is of abstracted functions rather than complex, low level hardware components<sup>3</sup>. The final level, emulation at the application level, requires building an emulator that mimics the exact behaviour of a computer program. This is can be difficult without access to the required documentation and results only in a single emulated application, rather than an emulated computer or OS that can run many different applications. To narrow the scope of experiments to be performed during the PA/5 work package the focus will be on hardware emulation/virtualisation by emulators or virtual machines written in software, which offer the best potential for effective preservation, while ensuring support for a wide range of emulated applications.

---

## 3 Previous experiences in emulation testing

Emulation testing has been performed within a number of sectors, but it has largely been conducted in an ad-hoc or experimental manner. The approach taken in this document will be guided where possible by this existing work. The following sections provide some background to this document and detail approaches that might be developed or utilized further by Planets. A priority will be given to approaches that have provided practical strategies for emulation testing and have that have already yielded useful results.

---

### 3.1 Enthusiast experience

Over the last decade, a great deal of effort has been expended by computer game enthusiasts in the development of arcade machine and home computer emulators. Most emulation testing originating in this sector has been somewhat ad-hoc, but there are some useful practices that can be noted.

Emulator developers typically rely on visual comparisons where specific programs or applications (typically computer games or graphical demos) are run on emulators to test the reproduction of particular aspects of content, look or feel. In some instances this may involve simply observing whether a particular application executes under emulation without crashing. In others, particular well known aspects of the tested application are observed. An example on the BBC Micro is the game Elite, which provides a particularly stringent test of an emulator's reproduction of timing and video display<sup>4</sup>. Most emulator developers employ a series of tests to examine specific elements of their emulators so they can identify errors and refine the emulation quality of their software. Organised, independent evaluations of comparable emulators are few in number, but it is recognised that "execution testing" provides a quick but useful way of gaining an impression of the relative qualities of a number of different emulators. Execution without crashing of a particular set of test applications (typically games, which push many aspects of emulation completeness) provides a useful test which is quick and simple to conduct. Particular games are often well

---

<sup>3</sup> See discussion of emulation of the George3 executive in "A blueprint for Representation Information in the OAIS model", Holdsworth, D and Sergeant, D, <http://www.personal.leeds.ac.uk/~ecldh/cedars/ieee00.html>

<sup>4</sup> See the last bullet point: "Elite (computer game)", Wikipedia, [http://en.wikipedia.org/wiki/Elite\\_%28computer\\_game%29#Trivia](http://en.wikipedia.org/wiki/Elite_%28computer_game%29#Trivia)

known by games enthusiasts for pushing particular aspects of emulation and such “execution tests” will often extend to quick observations of how accurate the reproduction of these particular aspects is. An example of a comparative emulation assessment provides some indicators of the potential in this area<sup>5</sup>. In the MSX example, a small number of general criteria are assessed.

---

### 3.2 User Testing

The Camileon Project utilised carefully monitored User Testing<sup>6</sup> in an effort to understand the preferences future users might have in using digital objects preserved via different strategies (such as migration<sup>7</sup> and emulation)<sup>8</sup>. Careful design of User Tests is crucial in ensuring that the assessment of relevant test criteria is not unduly influenced by other factors (eg. In Camileon tests, the users viewed a computer game in its original environment, under emulation and in a re-created or migrated form. Most users preferred the migrated form to the emulated form, despite several key losses in significant properties).

---

### 3.3 Visual testing/comparison of specific criteria

As part of their recent emulation project in 2006/2007, NANETH and KB-NL produced a document describing an exercise to evaluate and compare a number of x86 emulators<sup>9</sup>. This work focused on a visual comparison of the reproduction via emulation of specific criteria, by a number of testers. This offers a potentially useful, if labour intensive approach. Of particular relevance is the set of criteria used, and the technique used to assess and rate the criteria. NANETH and KB-NL used the term “Reference Workstation” to refer to an original platform running original software and digital objects which was utilised as a point of reference in performing comparisons between the representation of digital objects under emulation and the object as it was originally seen and used. This term will be utilised in this document where applicable.

---

## 4 Approach taken by Planets for emulation testing

There are no easy answers in developing an approach to emulation testing. Previous experiences of testing emulators are few and far between, and there does not appear to be an ideal approach to accurately and efficiently test emulators. This document describes the best (or perhaps more accurately the “least worst”) approaches that have been devised by the authors following a number of discussion and brainstorming sessions. The authors have attempted to take a pragmatic approach to the subject, describing what they hope will be test strategies that will yield useful results despite being based often on human judgement rather than any more precise analysis.

While the authors were able to draw on previous experiences from projects such as Camileon and the NANETH/KB Emulation Project, this document should be considered as a summary and analysis of previous work, accompanied by the thoughts of the authors on the most suitable way to proceed. It is expected that as practical experience in implementing the strategies outlined in

---

<sup>5</sup> “MSX Emulator Comparison”, MSX Resource Center Foundation, <http://www.msx.org/MSX-Emulator-Comparison.articlepage8.html>

<sup>6</sup> i.e. logging each action (such as mouse clicks, pauses, reactions) a user performs in a user testing laboratory, and then analyzing the results.

<sup>7</sup> Migration is typically considered to be the conversion of data from one format to another (eg. migrating a MS Word file to a PDF). Where the data includes software, a “migrated” or “re-created” version could be considered to be a new version of the software that has been re-written in a new language and/or on a new computer platform. This is discussed further in “Migration a Camileon discussion paper”, Wheatley, P, <http://www.ariadne.ac.uk/issue29/camileon/>

<sup>8</sup> “Emulation vs. Migration: Do Users Care?”, Margaret Hedstrom and Clifford Lampe, Camileon, <http://www.ohio.rlg.org/legacy/preserv/diginews/diginews5-6.html#feature1>

<sup>9</sup> “Test Results Document, Experiment results and findings” by Jeffrey van der Hoeven, produced as part of the KB-NANETH Emulation Project (not externally available).

this document is gained by Planets, the approaches described will be refined and developed (perhaps significantly). This practical testing is expected to be conducted by Planets during the next phases of the project in the second half of 2007 and the first half of 2008.

The development of in-depth emulation case studies in 2008 (see PA/5 project plan for the second 18 months) is expected to lead to a further comprehensive revision of this document, extending the testing scope to areas such as the configuration of emulators, and user issues such as the necessity for user documentation. The ultimate aim being to provide a framework for testing that not only enables evaluation of emulators, but goes much further in enabling evaluation of successful preservation via emulation.

## 5 Emulation uses

In order to consider how emulation testing might be performed, it is first useful to consider the specific purposes emulation might be applied to in a digital preservation context. This table provides an introduction to these purposes with some initial thoughts on the implications for emulation testing. The issues raised are explored in more detail later in this document. The categorisation of these purposes is aims to separate the different emulation uses and enable particular testing requirements unique to these uses to be identified.

Emulation ID	Description	Example	Notes on requirements for testing
1	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	Software historian runs a George3 emulator in order to view and use the worlds first Algol68 compiler.	Emulation needs to be of sufficient quality to reproduce all or some of the elements of the original environment. This is explored in more detail below. This is highly dependent on the content being emulated. For the Algol68 example, replicating the speed of execution and other timing idiosyncrasies would not be necessary. For a computer game or multimedia animation, accurately reproducing the speed of execution would be essential. Testing therefore needs to be carefully tailored depending on the likely end use of the emulation
2	Emulation of obsolete platform in order to run application software used to migrate data	Emulating a Windows 3.1 machine in order to run an early version of Autocad, enabling migration of data to later versions of Autocad.	Requires sufficient emulation quality to ensure a particular migration application can run, although there needs to be some confidence that it runs accurately. Exact reproduction of timing and execution speed is not important.
3	Use of emulation to verify the accuracy of an alternative preservation	Where a Reference Workstation is not available, this would provide an alternative	Requires emulation accuracy to be sufficient to render the migration source file. Although very similar to number 1 in



	<p>strategy (eg. Migration). The emulation enables rendering an object in its original environment so it can be compared with a migrated version of the same object. Thus providing an indication of whether significant properties have been lost in the migration process.</p>	<p>strategy to compare preserved digital content with the original digital object. The approach would emulate a Windows3.1 platform in order to view a Microsoft Works document. This would then be compared with a version of the document which had been migrated through successive versions to Word2003. The emulated version would therefore act as a reference point by which other preservation actions could be evaluated.</p>	<p>practical terms, this purpose suggests use by an expert who might verify or evaluate the performance of other strategies. Requirements for the end user of the emulator being tested may therefore be different, particularly in areas such as user friendliness and documentation.</p>
4	<p>To gain access to data on obsolete hardware by emulating the software/hardware that runs an obsolete peripheral (or tape, mag tape, etc) drive. This use is commonly referred to as an element of the process of "digital archaeology".</p>	<p>Emulation of an obsolete machine for a tape drive which no longer has a surviving computer to operate it.</p>	<p>This is likely to be a strategy pursued only as a last resort to rescue data on obsolete media. Emulation requirements are likely to be focused almost entirely on supporting the obsolete platform and peripheral, while enabling data to be extracted as a byte stream or "disk image". Requirements for testing will be oriented specifically on these areas.</p>

---

## 6 Purpose of emulation testing

The following list describes the areas in which emulation testing, as described in this document, may be useful within Planets:

1. Assessment of emulation tools in order to select the most appropriate for use within Planets (eg. To be “wrapped” and delivered as Planets services).<sup>10</sup>
2. The recording of metadata describing the strengths and weaknesses of selected emulation tools in the tool registry, in order to facilitate the Preservation Planning process.
3. Assessment of emulation tools in order to facilitate benchmarking and certification activities<sup>11</sup> (may be undertaken in the latter stages of Planets).

---

## 7 The Planets approach to emulation testing

Planets will apply a formal approach to emulation testing, following the standard Planets experiment workflow.

---

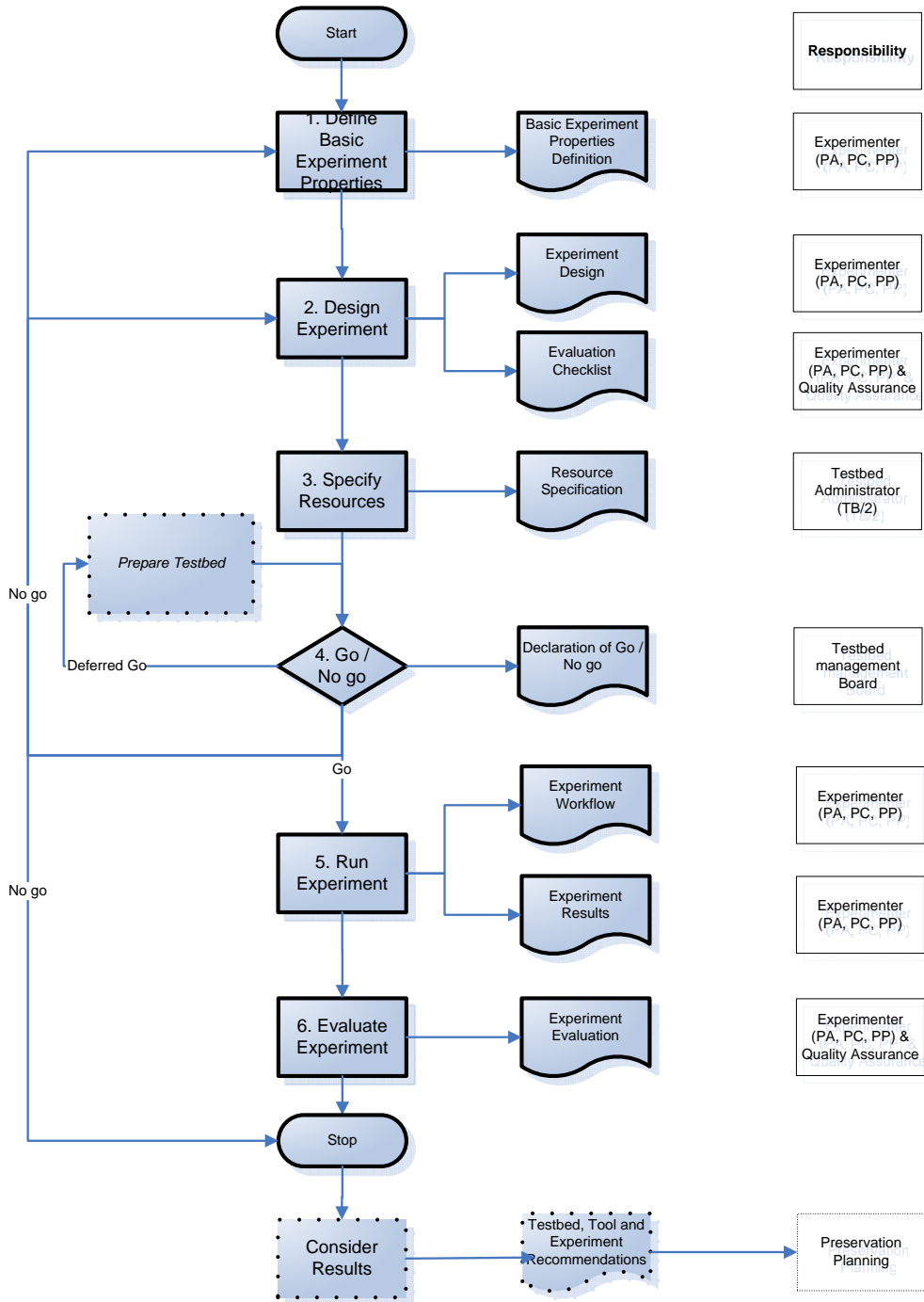
### 7.1 Workflow for emulation experiments

Emulation experiments will, where possible, follow the standard Planets workflow for performing experiments as defined in TB/3-D1 Testbed Workflow, Checklists and Templates:

---

<sup>10</sup> Initial Planets emulation testing will focus on tools that emulate selected environments. See the “Criteria for selecting environments” PA5 document.

<sup>11</sup> Given a particular digital object requiring preservation via emulation, it is currently a difficult task to select an appropriate emulator or virtualization tool for the purpose. Benchmarking of available emulators would provide evidence of the relative characteristics of particular emulation tools. Certification based on rigorous testing and evaluation might provide an independent indication of quality, longevity and appropriateness for preserving different kinds of digital objects.



The following list considers issues of particular consideration for emulation testing, with regard to the standard experiments workflow as outlined in the diagram above:

- Stage 1 includes the identification of appropriate Research Questions. Examples of questions relevant to emulation testing are provided below.
- Stage 2 includes identification of an appropriate Experiment Strategy as well as selection of appropriate Experiment Criteria (both described below).
- Stage 3 involves the selection and setup of appropriate resources, which might include emulators, operating systems, software and the digital objects themselves. Further consideration to technical issues affecting the setup and execution of experiments is described in Technical Preparation Issues, below.
- Stage 4 is the go or no go decision to proceed with the experiment.
- Stage 5 is the execution of the experiment itself. The experiment subjects are assessed and ratings allocated to the selected criterion.
- Stage 6 is the evaluation of the experiment, where conclusions are drawn based on the results of the experimental analysis.

---

## 7.2 The role of the Testbed in the execution of emulation experiments

Emulation testing may not be suitable for execution within the Planets Testbed due to a range of factors. Some tests may require use of resources external to the Testbed, such as a Reference Workstation. Some testing strategies are well suited to a Testbed environment where tests can be designed, scheduled and then left to run automatically within the Testbed, with the Experimenter returning hours or days later to view the results. Most, if not all, emulation testing requires almost constant input from a human tester or human test subject who "drives" the applications being emulated. Interaction of the user with the emulator and then feedback in terms of video and sound from emulator to user must be in real time. A remotely accessed Testbed may add lag or slow the response time which could severely bias the experiment.

At the time of authoring this document, it is unclear how emulators will be wrapped and integrated within the Planets Interoperability Framework. Indeed it may be impossible to do this. This again poses questions as to how the Testbed might be utilised during emulation testing as the emulators would not then be available for use within the Testbed environment as wrapped services.

Despite these issues it is clear that several functions of the Testbed will be useful to emulation testing even if the tests themselves are not suitable for execution within the Testbed environment. In particular, the recording of results in an experiment database, the formalised experiment workflow and design process, and the provision of a stable reference environment that enables experiments to be replicated. It therefore seems sensible to utilise the Testbed for emulation testing whenever practical. Ensuring realistic response and interactivity between user and test environment may necessitate that a local Testbed instance is used rather than a remote one. In a worst case scenario, emulation experiments would run outside the Testbed environment. This may be inevitable if wrapped emulation services are not available (as discussed above). However, in these cases the standard workflow should still be followed (omitting the processes linked directly to Testbed functions) and recorded results should be uploaded to the experiment database.

The process of wrapping emulators, how this might be done, and its implications for using and testing emulators will be explored elsewhere by Planets PA/5 work at a later date.

---

## 7.3 Research Questions

In order to enlarge the Planets Project's understanding of the uses of emulation or virtualisation for digital preservation purposes, so called 'Research Questions' will be formulated. Answering these Research Questions will provide us with insights into the key aspects of emulation or virtualisation in a digital preservation context. These Research Questions will guide our investigation and will continue to be adapted during the Planets project. New questions will be added and old ones refined. The research questions can also help us to define new experiments by raising issues that may need to be followed up with further investigation.

Each experiment will address one or more general Research Questions that are applicable to most experiments. Additional Research Questions might explore issues specific to a particular experiment. For example, specific questions may address a particular object type, an digital object attribute, an authenticity requirement, a technical issue or a specified combination of these. The Research Questions are embedded in the experiment process and made explicit in the experiment design.

Relevant Research Questions should be chosen in stage 1 of the standard experiment workflow. Possible Research Questions covering the key topics of interest to Planets emulation testing include:

### 7.3.1 Authenticity

*Is the emulation or virtualisation tool we have used for a number of experiments able to authentically represent (a selection of) digital objects?*

For archival institutions this is a key question: if the authenticity of the digital objects cannot be guaranteed, the emulation or virtualisation tool is not suitable for preservation purposes for this particular set of digital objects.

*“An authentic object is an object that can be proven*

- *to be what it purports to be*
- *to have been created or sent by the person purported to have created or sent it and*
- *to have been created or sent at the time purported*

*To ensure the authenticity of records (=digital objects), organisations should implement and document policies and procedures which control the creation, receipt, transmission, maintenance and disposition of records to ensure that records creators are authorized and identified and that records are protected against unauthorized addition, deletion, alteration, use and concealment.”<sup>12</sup>*

### 7.3.2 Strengths and weaknesses of emulation

*What are the strengths of the tested emulation or virtualisation tool compared to other tools we have tested so far?*

In order for Preservation Planning to make the right choices for a preservation strategy, we need to know which emulation or virtualisation tool is able to do the job for us, given a certain hardware environment, digital object, etc. Every emulation or virtualisation tool will have strengths (advantages) and weaknesses (disadvantages). It is important for us to record these advantages and disadvantages by answering this research question.

### 7.3.3 Effectiveness/appropriateness of a tool

*What are the factors that affect the effectiveness or appropriateness of a particular tool: type of digital object? authenticity requirements, supporting resources, e.g. staff, equipment?*

Some emulation or virtualisation tools are more effective or appropriate than others. What is influencing the effectiveness or appropriateness of a tool? For instance, the characteristics of the digital object which will run under emulation. It can be expected that for one type of digital object emulation or virtualisation is more effective/appropriate than for another type of digital object.

### 7.3.4 Metadata

*What set of descriptive metadata is required to fully describe emulation or virtualisation tools. What metadata is missing (if any) in order for others (within and outside PLANETS) to be able to grasp the value/merits of a particular emulation or virtualisation tool?*

By performing experiments we will acquire more knowledge about the different tested emulation and virtualisation tools and the essential characteristics of them. This might urge us to change our list of descriptive metadata fields.

---

<sup>12</sup> ISO 15489-1 Information and documentation – Records management Part; 1 General; 7.2.2 Authenticity.

### 7.3.5 Technical issues

*What technical issues play a role with the tested emulation or virtualisation tool? What needs to be resolved technically in order for these tools to play a role in digital preservation?*

If (cultural heritage) organisations want to use emulation or virtualisation tools in order to preserve their digital objects, there will always be technical issues to solve, such as the required image format of the application software or copyright/software license issues. By making these technical issues (more) explicit we know what is required to actually use these emulation or virtualisation tools in a digital preservation context.

---

## 7.4 Costs

*What does it cost to apply emulation in this particular case (for example: this environment, these digital objects, these authenticity requirements)?*

The cost of the preservation solution is an important criteria in preservation planning.

---

## 7.5 Experiment Strategies

The Experiment Strategy is the approach used to test or evaluate the target of an experiment. This should be chosen in stage 2 of the standard experiment workflow. The strategies are:

1. Visual Comparison
2. Automated Comparison
3. User Testing
4. Benchmarking
5. Execution Testing

These are described in more detail below.

### 7.5.1 Visual Comparison

This approach involves a detailed human comparison and evaluation of specific criteria experienced by the tester on the Reference Workstation and the emulated one. Each tester would compare the reproduction of the specific criteria under emulation with its representation running on the Reference Workstation, recording a rating indicating the quality of reproduction and any notes describing particular aspects or issues that may be of relevance.

This strategy has been employed with some success by the NANETH/KB Emulation Project described above, and mirrors the detailed visual assessments performed by the Digital Preservation Testbed Project<sup>13</sup> in their tests of migration strategies. Visual Comparison is expected to form a crucial role in emulation testing performed by Planets, mainly due to the shortcomings of other approaches. Despite this, it should be noted that Visual Comparison is time consuming, laborious and is therefore likely to be expensive. Results are dependent upon the judgement of the testers, and so accuracy will depend not only on careful design of such experiments, but also on the number (and background) of testers involved. It is hoped that practical experience of such testing will provide some indication of the minimum number of testers required to achieve a reliable and sufficiently accurate result.

### 7.5.2 Automated Comparison

Automated Comparison is the strategy of performing a comparison of specific criteria, as in the Visual Comparison approach, but utilising an automated method of assessment. An example might be the capture of key frames during the execution of specific software. Key frames captured on the Reference Workstation and under emulation would then be compared for differences between the frames<sup>14</sup>.

Planets is unaware of any available facilities for performing Automated Comparison and a number of technical challenges are yet to be faced. To pursue this strategy, tools and approaches would need to be devised and implemented but a number of problems remain to be solved. While

---

<sup>13</sup> Digitale Bewaring, <http://www.digitaleduurzaamheid.nl/index.cfm?paginakeuze=286&lang=en>

<sup>14</sup> Often referred to as “diffing” two files, <http://en.wikipedia.org/wiki/Diff>

it may be possible to build new emulators with facilities to assist Automated Comparison, this approach may not be possible with existing emulators. It remains unclear how (for example) key frames might be captured on the Reference Workstation. Another key difficulty is in ensuring that user input could be repeated identically for each emulator of Reference Workstation involved in an experiment.

If Automated Comparison can become a viable approach through Planets developments, it might provide a less subjective assessment of criteria than Visual Comparison without requiring a large amount of human assessment effort. However, even in a best case scenario it seems likely that the range of criteria that this approach might provide useful information on is limited.

### 7.5.3 User Testing

User Testing involves the observation of a number of users while performing a series of tasks on emulators and/or Reference Workstations. The test might involve the capture of a user's performance, reactions, or opinions on their experience.

Evaluation can be performed via a number of methods:

1. Recording and analysing a user's actions (e.g. their movements, mouse clicks, key presses)
2. Assessing a user's performance in successfully completing the tasks set
3. Assessing a user's opinions or satisfaction via questionnaire or interview

The CAMiLEON User Testing, described above, set tasks for users to complete on a Reference Workstation running a particular computer game and on an emulated and migrated version of the same game. The test required users to reach level 3 of the game. The users actions were captured on video and analysed afterwards. Interviews with the users provided information on satisfaction and revealed any preferences for the alternative versions.

American students were specifically chosen as neutral test subjects to play a British computer game from the 1980s<sup>15</sup>. The test designers hoped that this would ensure a balanced view between alternative versions of the game, allowing preferences between the original, migration and emulation to be captured. In performing the tests the experiment designers were amazed at how much the subjects struggled to complete what was for them an easy task. To the designers it was a simple platform game, the like of which they had grown up with. The subjects had grown up in a world of 3D computer games and lacked the basic skills to play a 2D platform game! Strong preference was shown by the users for the migrated (or re-created) version of the game, despite the loss of some key significant properties in this version. It was unclear as to whether the users were even aware of this loss, or simply preferred what appeared to be more modern and easy to use software. This example shows how hard it can be to design effective experiments to elicit information on specific criterion, when complex and often subtle issues resulting from user experience (or lack of user experience) need to be considered.

User Testing offers the ability to capture information which cannot be obtained from another source. Potentially it can provide valuable insights into user preferences which may prove invaluable in distinguishing between the reproduction of essential or non-essential emulation criteria. It is however a time consuming and complex process. Methodical design of the experiments and careful analysis of results is required in order to achieve a reliable and useful conclusion. Performing experiments with a sufficient number of users can require a great deal of effort. Assessing what a representative sample is, can in itself, be a difficult task.

### 7.5.4 Benchmarking

This approach involves the execution of benchmarking software that analyses specific aspects of the environment it is executing within, and provides measurements of criteria such as performance.

This approach is expected to be relatively simple and cost effective to implement but will only provide a limited range of useful information. Interpretation of the results also requires caution. For example, benchmark tests providing information on speed of execution might need careful analysis. Slow execution may be considered a problem if video or animation is likely to be

---

<sup>15</sup> Chuckie Egg, a "classic" arcade platform game from the 1980s, [http://en.wikipedia.org/wiki/Chuckie\\_Egg](http://en.wikipedia.org/wiki/Chuckie_Egg)

emulated, but in the longer term, as the hardware that the emulator runs on improves, this might become less of a concern. A result which shows a much higher performance than the Reference Workstation might be considered positive for some uses where much processing is required before a useful result is returned, or a hindrance where the timing of animation or other aspects makes the emulator difficult to use.

### 7.5.5 Execution Testing

This strategy involves the execution of a set of example applications on an emulator or set of emulators, to assess which result in successful execution and which result in obvious crashes. This approach could provide a high level assessment of emulation quality, perhaps utilising a scale as used in the MSX test described above under “Previous experiences in emulation testing”.

This strategy is similar in principle to Visual Comparison, but is considered as a separate approach as it focuses on a shorter and less detailed evaluation at a higher level. This technique may be suited to providing a general impression of emulator quality, which can be backed up with more detailed Visual Comparison tests where necessary. For example when assessing which available emulators for a particular platform might be worthwhile adopting, a first pass could perform Execution Tests to narrow the field, before the more time consuming and detailed Visual Comparison test are performed.

---

## 7.6 Experiment Criteria

The following table describes the criteria which might be assessed during an experiment<sup>16</sup>. Experiment criteria should be defined during stage 2 of the standard experiment workflow.

Criteria names	Definition
<b>Content</b>	
Text	The numerical and alphanumerical characters, including special characters like !, ", é, ë, and national special characters which present a particular challenge in a pre-Unicode environment.
Images	Graphical still representations in the content, illustrations, pictures, schemas, 3D-models.
Animation	Graphical moving representations in the content, like dynamic illustrations, dynamic 3D-models, movies.
Sound	Intended “noise” like speech, music or signals.
<b>Structure</b>	
Internal structure of the object	Organisation of the elements within the object. Examples: sections in an e-book, XML schema of an XML document, rows and columns of a spreadsheet.
Cohesion between object and environment	Dependencies of the object on its environment. Examples: viewer applications, fonts in a text document, codecs of a movie, dll's for program functionality.
<b>Context</b>	
Meaning of the object	The actual intension of the object. Examples: meaning of a single record in a range of objects that form a collection. Metadata on bibliographical, technical and preservation level.
<b>Appearance</b>	

<sup>16</sup> This list has been developed from a criteria list described in “Test Results Document, Experiment results and findings” by Jeffrey van der Hoeven, produced as part of the KB-NANETH Emulation Project.



Text format	The layout of the text in an object. Examples: font face, italic, bold, columns, margins.
Colours, contrast, brightness	The colouring of the object on screen. Examples: monochrome screen, 16-bit colours, low contrast.
Screen resolution	The number of dots (pixels) that are used to represent the object on visual output. Examples: 800x600, 1024x768.
Screen aspect ratio	The proportions of an object on screen. Examples: 4:3 monitor versus 2:1 widescreen.
Frame rate / speed of execution	The performance of the rendering. Examples: screen update of 25 frames per second.
Screen refreshment	The frequency of the visual output. Examples: 60Hz, 85Hz, 120Hz.
Smoothness of animation	The way a computer handles the transition from one visual frame to another.
Smoothness of sound	Constant sound reproduction.
Sound volume	The intensity of the sound.
Sound quality	The purity of the sound.
<b>Behaviour</b>	
Performance	The computation power of the computer.
User interactivity	The way the computer response on user input and the ability to give feedback.
Mouse behaviour	The mouse response to user input.
Keyboard behaviour	The keyboard response to user input.
Software behaviour	The correct reproduction of the functionality of the software

Each of the criteria will be rated using the following scoring model:

Rating	Explanation
+	accurate / works correctly
+/-	moderate / works partly
-	poor / does not work
n/a	not applicable to this object

In addition to this rating, comments can be added to each criterion to provide additional detail and explanation. For example: the criterion “colours, contrast, brightness” has been rated with “+/-” because the colour red seems to be different compared with the Reference Workstation. These ratings were used with some success in the KB-NL / NANETH emulation experiments mentioned above. It is expected that further experience of performing testing will provide an indication as to the appropriateness of this scale and the need for clear guidance in its use. Assessment and allocation of ratings is subjective so a pre-agreed approach in this area is essential.

Some criteria are likely to be considered critical and some non-critical depending on the subject of the experiment and the ultimate use of the emulation for which the process of testing has been initiated to inform. For example, when emulating to enable use of a migration tool (emulation use 2) smoothness of animation and frame rate are unlikely to be critical. When emulating to enable the use of an interactive multimedia program, those criteria would be critical.

Experiments should therefore either focus exclusively on critical criteria, or apply a method weighting to critical and non-critical criteria.

The following table illustrates which Test Strategies will yield information on which Test Criteria. A “Y” with an orange fill indicates good coverage, and “P” with yellow fill indicates partial coverage. Further experience in assessing criteria in actual experiments will enable the refinement of this table in future revisions of this document.

Criteria	Visual Comparison	Automated Comparison	User Testing	Benchmarking	Execution Testing
<b>Content</b>					
Text	Y	Y			P
Images	Y	P			P
Animation	Y				P
Sound	Y	Y			P
Structure	Y				P
Internal structure of the object	Y	P			P
Cohesion between object and environment	Y				P
<b>Context</b>					
Meaning of the object	Y		Y		P
<b>Appearance</b>					
Text format	Y	P			
Colours, contrast, brightness	Y	Y		P	
Screen resolution	P	Y		P	
Screen aspect ratio	P	Y		P	
Frame rate / speed of execution	P			Y	P
Screen refreshment	Y			P	P
Smoothness of animation	Y				P
Smoothness of	Y				P

sound					
Sound volume	Y	Y			P
Sound quality	Y	Y		P	P
<b>Behaviour</b>					
Performance	Y		Y	P	
User interactivity	Y		Y		
Mouse behaviour	Y		Y		
Keyboard behaviour	Y		Y		
Software behaviour	Y		Y		P
Data extraction	Y				

## 8 Risks and dependencies

### 8.1 Technical preparation issues

This section provides a brief overview of technical issues which might need to be considered in the design and execution of emulation experiments.

To be able to perform experiments, the emulation/virtualisation process must be installed and configured. As each emulator or virtualisation application is different, preparing an experiment is not a standard procedure. The following issues should be considered:

1. An emulator or virtualisation tool is typically built to run on a particular host platform (technical environment). Although different host platforms can be supported, this results in different binary installation files that are suited for one particular platform each. It is important to know which host platform is used and if it is supported by the emulator or virtualisation tool.
2. Aside from the emulator, software like the operating system and additional software applications are required. Further dependencies of the object (to be emulated) on the target environment could be: special fonts, codec's or dll's.
3. All required software that will be used by the emulator has to be packaged in so called disk images. A disk image is a (large) file containing an exact copy of a file system of a physical (e.g. floppy) or logical (hard disk partition) carrier. Although disk images are quite common today, no real standard on file formats is available, except from CD-ROM/DVD-ROM (can be translated into ISO-files). This has led to various implementations for different emulators.
4. To setup an emulation process to the needs of the user, configuration of the emulator or virtualisation tool has to be performed. Depending on the tool, configuration occurs in one of three different ways:
  - a. Using a graphical user interface
  - b. By using a configuration file (txt)
  - c. Command line parameters

### 8.2 Timeliness

Access to the Reference Workstations is likely to be critical for most forms of emulation testing as is highlighted above. Of course, as technology evolves, older computers and their respective

operating systems and software are not always readily available. The CAMiLEON<sup>17</sup> Project highlighted the importance of taking action to implement emulation strategies while access to the original computers, software and human knowledge and experience of them was still available. This is certainly a key risk to emulation activities, and one which will impact on testing in particular.

Fortunately activities performed by both enthusiasts and professionals in the cultural sector may extend the window of opportunity for action involving emulation to be taken. A range of enthusiasts, institutions involved in preservation activities such as The National Archives, and cultural institutions such as Bletchley Park, collect obsolete documentation, software and hardware. A surprising number of Computer Museums<sup>18</sup> can be found, and interest in the history of computing and retro computer gaming remains strong<sup>19</sup>.

It is currently unclear as to whether this issue will impact significantly on the work of Planets, and this will depend largely on the selection of target platforms for emulation. A focus on older and less accessible machines may require effort to locate, acquire and utilise original obsolete computers and related software. Partnership with those involved with collecting the physical history of computing, as listed above, is likely to be the most effective strategy.

---

## 9 Emulation experiment example: Educational multimedia application

This section provides an example of how an experiment might be designed and executed. As Planets moves on to perform emulation testing in practice, it is expected that this section will be populated with a number of real life experiment examples.

The experiment will focus on testing an old educational multimedia application that is part of the collection of interactive multimedia publications stored in the KB-NL.

The aim of this experiment is to test if this application will run on an emulated environment and how accurately it will execute in comparison to its execution on the reference workstation. We would like to know if there are any major differences between them. If so, are those differences problematic for the execution and use of the application?

This experiment will utilise the Visual Comparison approach. As we would like to perform an experiment in the way it is described under number 1 of emulation uses, the best way to do so is performing a visual comparison. This strategy allows us to compare all (or most) of the criteria between the Reference Workstation and emulated environment. However, it also means that this approach will be labour intensive and requires access to a Reference Workstation. The experiment will evaluate all the criteria listed earlier in this document.

The multimedia application was published in 1999 and requires an MS Windows 9x computer environment with at least 64 MB of physical memory. The application needs to be installed before it can run. Moreover, it requires the original CD-ROM (physical or emulated) in the drive during execution.

To perform this experiment using the Planets Test-bed, the most practical approach is to run the original and emulated environment locally. This means that an original PC from around 1999 should be used with MS Windows 95/98 installed (the Reference Workstation). Right next to it, an emulated environment should be created. This requires a modern (fast) computer with an emulator installed on it which is capable of mimicking the IBM PC or compatible architecture (probably Intel Pentium, but at least Intel 486 DX/2 with 64 MB of internal memory) running MS Windows 95 or 98. Furthermore, it should support a virtual CD-ROM drive together with a disk

---

The following table describes the criteria which might be assessed during an experiment<sup>17</sup>. Experiment criteria should be defined during stage 2 of the standard experiment workflow.

<sup>17</sup> "Emulation, Preservation and Abstraction", Holdsworth, D and Wheatley, P, <http://www.rlg.org/preserv/diginews/diginews5-4.html#feature2>

<sup>18</sup> [http://en.wikipedia.org/wiki/Computer\\_museum](http://en.wikipedia.org/wiki/Computer_museum)

<sup>19</sup> In the UK, the presence of the monthly "Retro Gamer" magazine illustrates the popularity of retro gaming in the context of a growing computer games and entertainment market <http://www.retrogamer.net/>

image of the original CD-ROM, input devices like keyboard and mouse and a screen as output device.

To what extent the test-bed can support this is still unclear. Most likely, the experiment itself will be executed locally. This means that the Reference Workstation and the emulated environment both needs to be installed and configured outside the test-bed. However, the experiment will be done using the same procedures as defined by the test-bed. The experiment results in one or more documents which can be preserved in the test-bed environment.

---

## 10 Conclusions

Emulation testing is an area which has not previously been explored in detail. Previous work has scratched the surface of this problem but does not provide all the answers. This document provides a starting point for the development of successful approaches to emulation testing, but the next phase of work must depend on practical experience and experimentation. 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. Now that the initial thoughts of PA/5 have been discussed and captured in this document, greater interactions with the other sub-projects will now be beneficial.

The following recommendations are made for future activities within Planets:

1. Construct emulation case studies using digital content from real life collections provided by the Planets partners.
2. Collaborate with the other PA work-packages and related sub-projects. This would enable PA work-packages to reach consensus on criteria for testing, and the description of PA tools. Collaboration with PP and PC work-packages may enable testing activities to be focused on different categories of digital objects under emulation. This may assist with selection of appropriate emulators for specific tasks.
3. Perform emulation testing on emulators developed within Planets and emulators available from other sources. This will move us toward capture of the knowledge required to inform the appropriate selection of emulation strategies and specific emulators as part of the preservation planning process.
4. Refine and update this document based on practical experiences.