Project Number	IST-2006-033789
Project Title	PLANETS
Title of Deliverable	Test methods for Testbed
Deliverable Number	D2
Contributing Sub-project and Work-package	TB/3
Deliverable Dissemination Level	Internal PU
Deliverable Nature	Report
Contractual Delivery Date	30th November 2007
Actual Delivery Date	30th November 2007
Author(s)	NANETH

Contributors

Person	Role	Partner	Contribution
Petra Helwig	TB/3 partcipant	NANETH	Author
Judith Rog	TB/3 partcipant	KB-NL	Review
Caroline van Wijk	PA/3 lead	KB-NL	Review
Eleonora Nicchiarelli	TB/3 participant	ONB	Review
Manfred Thaller	PC/2, PC/4 lead	UzK	Review

TABLE OF CONTENTS

1 Introduction

1.1 Context

As stated in the PLANETS Description of Work for work package TB/3, task TB/3.2 will

[…] develop test methods, contributing to the quality assurance. It will refine and adjust validation methods for the required experiments on digital objects. These validation methods will, in the final release, be aligned with the validation / evaluation model developed in the Preservation Planning sub-project (PP/5).

This deliverable is the result of this task.

1.2 Purpose of this document

Summary:

Main purpose:

- *Describe which test criteria can be defined in Testbed experiments.*
- *Describe how to evaluate the tests.*
- *Describe how to use the results.*

The Testbed is an application that will be used to carry out tests on tools for objects and tools for environments as developed and / or inventoried by the PLANETS sub-projects Preservation Action (PA) and Preservation Characterisation (PC). The main purpose of this document is to describe in detail the kind of tests that can be carried out on these tools with the Testbed, and the kind of results that will be produced by these tests. It will also give suggestions for the way the Testbed results can be used by other sub-projects and work packages.

This document will focus on the methods that will be used in the Testbed for testing tools for objects and tools for environments. It is not about software testing the testbed-applicarion itself, which will be done in TB/2.

1.3 Terminology

Summary:

"Tools" should be read as "services".

Strictly speaking, the Testbed will not provide the means to test *tools*, but *services*, which are wrapped tools. One tool could even be wrapped into more than one service, using e.g. different parameters. Therefore, where this document mentions a *tool*, this should be read as a *service*.

1.4 Relation to other documents

Summary:

This deliverable extends the Workflow and Checklist Document, specifically step 2. Design Experiment and step 6. Evaluate experiment.

The functionality of the Testbed, from a user perspective, is described in the *Use Case Document*.

The statements in the Use Case Document form the basis of the description of the functionality of the Testbed, as elaborated in the *Software Requirements Document*, which describes the system from a software perspective rather than a user perspective, still focussing on *what* the system should do, rather than *how* it should do it. The latter is described in the *Testbed Design Document*, which will mainly be used by the software developers.

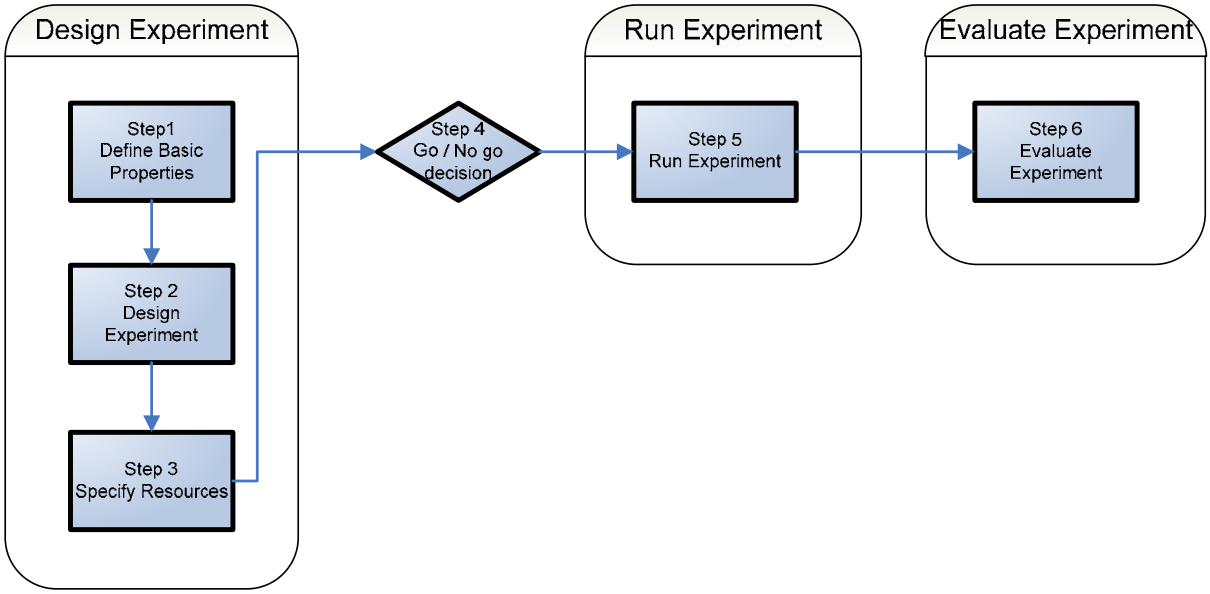
The core functionality of the Testbed is to enable a user to design, carry out and evaluate an experiment on certain data with a certain registered tool. The way this is done is described in the *Workflow and Checklists Document*, where the experiment process that is used for all experiments is described.

Every experiment is carried out according to a fixed experiment process, or workflow, that consists of six steps:

1. Design the experiment: specify what the experiment is about and what you are going to test: steps 1-3.
2. Carry out (run) the experiment: step 5.
3. Evaluate the experiment: step 6.

Between design and execution of the experiment, a go / no go decision is taken: step 4.

Each of the six steps corresponds with a web page, where the user fills in the necessary fields.



Part of step 2: Design Experiment, is to define the evaluation criteria, i.e. to specify what exactly is going to be tested. The evaluation criteria will be used in step 6. Evaluate Experiment, to decide to what extent the experiment has been successful. The *Workflow and Checklists Document* states that a list of evaluation criteria should be compiled, but does not state exactly how this should be done or which criteria could be used.

The purpose of this document is to give detailed information about which evaluation criteria can be used in the various experiment and also to zoom in on the way the experiments should be evaluated.

2 Purpose and Role of the Testbed

2.1 What do we need for digital preservation?

Summary:

We need tools, and a plan that states which tools to use in which situation.

To preserve digital objects, we need

- Preservation tools, e.g.
  - Characterisation tools: to extract essential characteristics of the digital object from a file.
  - Migration tools: to migrate digital objects, stored in an older or obsolete file format, to another format.
  - Emulation tools, to render digital objects in their original context on a new infrastructure.
- A preservation plan. A preservation plan indicates which preservation actions / tools will be applied to a collection of digital objects. A preservation plan is specific for an organisation and a type of objects, and depends on a.o. the preservation policy and guidelines of the involved organisation, its collection, its users and usage of objects etc.

2.2 Need for structured evaluation criteria

Summary:

Structured evaluation criteria are essential for searching and comparing the Testbed results.

There are several ways of evaluating an experiment. One possibility would be to simply write down any remarks in a free-format way. In the Testbed application this would mean, providing one or more free text field where the user can enter any comments on the experiment, or have a possibility to upload a (pdf or word) document.

The problem with this approach would be that it would result in lots of information on experiments in a non-structured way. A major goal of the Testbed is to make sure the results of the experiments are comparable and searchable. We would like to have the results in such a way that users will be able to ask questions (i.e. search the Testbed results) like

- Which migration tools are available for migrating images that preserve the colour correctly?
- Is there a characterisation tool that can extract the values for font family and variant?

Therefore, we need to structure the evaluation criteria.

2.3 Purpose of the Testbed results

Summary:

The Testbed results should provide information on the usability of various preservation tools, for various types of digital objects.

It is important to remark that we start from the assumption that the quality of a tool is dependent on the type of digital object it is used on. For instance, the fact that a certain migration tool works “very well”, whatever that may mean, on text-files does not necessarily mean that it also works well on images.

In the Testbed, various types of preservation tools will be tested, on several types of digital objects. Every experiment is carried out with one tool, on one type of digital object. The result of the Testbed experiments should be to produce information on how well the preservation tools perform on different types of digital objects.

Ideally, at the end of PLANETS, we hopefully have Testbed results for many tools, on many types of digital objects, as is shown in the following diagram:

Tools → Digital objects ↓	Tool 1	Tool 2	Tool n
Digital object type 1	*Results experiment 1* *Results experiment 5* *Results experiment 6*	*Results experiment 2* *Results experiment 7*	*Results experiment 3* *Results experiment 8*
Digital object type 2	*Results experiment 4* *Results experiment 11*	*Results experiment 10* *Results experiment 12* *Results experiment 15*	*Results experiment 9* *Results experiment 13*
Digital object type 3	*Results experiment 18* *Results experiment 20* *Results experiment 21* *Results experiment 22*	*Results experiment 16* *Results experiment 17*	*Results experiment 14* *Results experiment 19*
Digital object type n	*Results experiment 23* *Results experiment 24*	*Results experiment 25* *Results experiment 28*	*Results experiment 26* *Results experiment 27* *Results experiment 29* *Results experiment 30*

In detail: one cell of the above table would contain the results for one tool on one digital object type. For instance, if we consider say the experiments carried out with tool 1 tested on a digital object of type 3, the results will contain the following information:

Tool 1, Digital object type 3	Criterion 1	Criterion 2	Criterion n
Experiment 18	*Evaluation value*	*Evaluation value*	*Evaluation value*
Experiment 20	*Evaluation value*	*Evaluation value*	*Evaluation value*
Experiment 21	*Evaluation value*	*Evaluation value*	*Evaluation value*
Experiment 22	*Evaluation value*	*Evaluation value*	*Evaluation value*

2.4 Which types of digital objects?

Summary:

Distinguish digital object types according to their content, so e.g. text, image, audio, video, database, application, …

Firstly, we should decide which types of digital object we should distinguish. Generally speaking, there are two ways of doing this:

1. Distinguish according to the content. This would lead to digital object types like text, image, sound, database etc.
   Example: a scan of a letter, saved as .bmp would not be considered an image but text, because of its content.
2. Distinguish according to file format. This would lead to digital object types like .pdf-files, .doc-files, img-files etc.

Option 1 seems the more logical solution, because then we can focus on questions like e.g.: “How well is the font size preserved if we use tool X to migrate a text file from a .doc-format to a .pdf-format?”. So, the digital object type in this example would be: text file.

N.B. A slight difficulty, but not insurmountable, in this approach is that a single object often contains a combination of different types of content, e.g. a document that contains both text and images, or a website or multi-media application containing text, images, sound and video.

Note: the final list of digital object types has yet to be compiled. How this should be done and which work package should come up with such a list, is also to be decided. Probably, the easiest solution would be to use the classification used by or compiled by sub-project Preservation Planning (PP).

2.5 Which evaluation criteria?

Summary:

For characterisation experiments, migration experiments and emulation experiments, the evaluation criteria can be based upon the properties of a digital object type.

Secondly, we need to come up with a set of possible evaluation criteria of each digital object type.

2.5.1 Testing of functionality

The most important goal of testing preservation tools, is to assess their functionality. In other words:

- For characterisation tools, we will want to assess how correctly the tool extracts intellectual characteristics of the digital object from a file.
- For migration tools, we will want to assess how correctly the tool migrates digital objects, stored in an older or obsolete or otherwise undesired file format, to another format.
- For emulation tools, we will want to assess how correctly the tool renders digital objects in their original context in a new infrastructure.

For migration and emulation experiments, we can specify this as follows:

- For migration tools, we will want to assess if the intellectual characteristics, or properties, of a digital object are unchanged after the migration.
- For emulation tools, we will want to assess if the intellectual characteristics, or properties, of a digital object are unchanged in the emulated old environment.

In other words:

- For characterisation experiments: test to what extent the values of the appropriate elements of content, context, appearance, structure and behaviour of test objects, as characterised by the characterisation tool, are the same as the “actual” values.
- For migration experiments: test to what extent the values of content, context, appearance, structure and behaviour of the migrated digital objects are the same as the values of the original objects, by using the particular migration tool.
- For emulation experiments: test to what extent the values of content, context, appearance, structure and behaviour of digital objects in the environment emulated by the emulation tool, are the same as the values of the objects in their original environment.

So, we will need a list of all intellectual characteristics, or properties, per digital object type. This list should contain properties that have to do with the “intellectual” qualities of a digital object type and *not* with the file format. For instance, file size would not be an intellectual property because it could change when migrating to another file format. Nor would “resolution” be a good intellectual property of an image, because it only has meaning for images stored in formats like jpeg, bitmap etc. but not for vector images. Instead, the appropriate property would be “size of smallest detail”.

Example:

Imagine, we have a digital object type called “text” with property “font style” and “number of pages”.

- Characterisation experiments will test if the characterisation tool manages to characterise the appropriate properties well, that is, to what extent the values as characterised by the tool are the same as the “actual” values of the object properties. The evaluation criteria will therefore be: “Correct characterisation of…” + <Digital Object property>, or, more precise: “Similarity of characterised value and reference value of… + <Digital Object property>, e.g.
    - Similarity of characterised value and reference value of font style
    - Similarity of characterised value and reference value of page numbering

- Migration experiments will test if the appropriate properties for the test object at hand are preserved well, using a particular migration tool, therefore the criteria will be: "Correct preservation of…" + <Digital Object property>, or, more precise: "Similarity of value of … + <Digital Object property> + in original and migrated object", e.g.
  - Similarity of value of font style in original and migrated object
  - Similarity of page numbering in original and migrated object
- Emulation experiments will test if the appropriate properties for the test object at hand are rendered well, using a particular emulation tool, therefore the criteria will be: "Correct rendering of…" + <Digital Object property>, or, more precise: "Similarity of value of … + <Digital Object property> + of object in original and emulated environment", e.g.
  - Similarity of value of font style of object in original and emulated environment Rendering of page numbering

After some experiments, we would have Testbed results like, for example:

Characterisation tool X	***Characterisation of Width***	***Characterisation of Height***	***Characterisation of Colour depth***	***Characterisation of Resolution***
Digital Object Type: Image				
Experiment 1	quite similar	quite similar	quite similar	exactly the same
Experiment 2			quite similar	quite similar
Experiment 3	exactly the same	exactly the same	very different	very different
Experiment 4	quite similar	quite similar		
Experiment 5	exactly the same	exactly the same	exactly the same	exactly the same

and

Migration tool Y	***Migration of Font type***	***Migration of Font Size***	***Migration of Headers***	***…***
Ditigal Object Type: Text				
Experiment 1	quite similar	quite similar		
Experiment 2	exactly the same	exactly the same	exactly the same	exactly the same
Experiment 3			exactly the same	completely different
Experiment 4	quite similar	quite similar		

N.B. 1 The scale and meaning of these qualifications will be discussed in paragraph 4.2 Evaluation criteria.

N.B. 2. Since tools do not always perform the same for all file formats that they can read and write, it is important to keep track of the file formats that were used in the experiments, so be able to show the results for a tool on a certain digital object type, using a certain file formats. This would give us, for example, the above table for Migration tool Y, used on a digital object of type Text, when migrating from Microsoft Word (.doc) format to Open Document Format (.odf) format.

2.5.2 Standard environment and testing of non-functional aspects

Summary:

Some information on non-functional aspects of services is also captured. However, these aspects are not "evaluation criteria" in the strict sense.

Of course, we can also want to test non-functional aspects of a tool. We must keep in mind that in the Testbed, the user will not interact directly with the tool at hand. The tools will be wrapped into services that will be called via e.g. a command line in the background.

Many lists exist that enumerate non-functional aspects of software. ISO 9126 is an international standard for the evaluation of software quality. The quality model established in the first part of the standard, ISO 9126-1, classifies software quality in a structured set of characteristics and sub-characteristics as follows:

- **Functionality** - A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs:
  - Suitability
  - Accuracy
  - Interoperability
  - Compliance
  - Security
- **Reliability** - A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time:
  - Maturity
  - Recoverability
  - Fault Tolerance
- **Ease of use** - A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users:
  - Learnability
  - Understandability
  - Operability
- **Efficiency** - A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions:
  - Time Behaviour
  - Resource Behaviour
- **Maintainability** - A set of attributes that bear on the effort needed to make specified modifications:
  - Stability
  - Analyzability
  - Changeability
  - Testability
- **Portability** - A set of attributes that bear on the ability of software to be transferred from one environment to another:
  - Installability
  - Replaceability
  - Adaptability

Since, as explained above, we are actually testing services rather than tools, we can gather information on only a limited set of the above listed aspects. All aspects that are about direct interaction with the tool, such as learnability, but also e.g. installability, cannot be tested with the Testbed.

We will, however, capture a certain set of data on experiments that will give information on some of the above mentioned aspects. The exact set of data to be captured has yet to de defined, but we an think of aspects like

- Speed: total running time of experiment, run time per step in case of a service that executes several operations after another (a so called workflow experiment)
- Use of resources e.g. network load

The Testbed will be an application that can be downloaded and run in an institution's own environment. Many of the above non-functional aspects, the outcomes might be influenced by the technical environment and platform the application runs in: different institutions that download the Testbed into different environments, could get different outcomes. Therefore, a central instance of the Testbed will be installed in an environment with known characteristics, to ensure that outcomes of non-functional aspects, captured in this environment, are comparable.

The data on non-functional aspects that is captured during the experiment are not evaluation criteria in the strict sense. They are not benchmarked against any standard and the results do not

have to be graded. The values will just be captured automatically and be made available for a decision support tool.

2.6 Testbed results should be institution independent

Summary:

The Testbed results should be as objective (i.e. not subjective) as possible, in order to be generally usable. They should not depend on the institution in question.

A PLANETS-wide Testbed, where each partner can see and build on previous experiments, gains from objective results. That is, Testbed results should not be influenced by circumstances or characteristics that are specific for the institution that carries out the experiment. Testbed results should add to the general knowledge of how various preservation tools perform on different digital object types, but do not depend on the institution that executes the experiments.

Example:

If a certain experiment results in the conclusion that, using a certain migration tool, the height of the migrated image is “quite similar” to the height of the original image, when migrating an image from a .bmp to a .jpeg-file, the assessment “quite similar” should be objectively measurable. It should be avoided that, for instance, institution A rates the quality of migration “quite similar” and institution rates it “very different”, because they use different ways of evaluation.

For instance, imagine a text object with font size 12pt. A migration tool X migrates this document to another file format, and in this process changes the font size to 11pt. Institution A concludes that the image heights of the original and migrated image are “quite similar” – ideally, this comparing and assessing a measure of similarity is done automatically. Institution B agrees with this assessment: the Tested result is usable for B, too.

However, when it comes to deciding which tools are usable in practice, for institution A migration tool X might be good enough – because the text only has to be readable – but this might be totally unacceptable for institution B. In other words, A and B will assign a different importance to the exact preservation of the font size.

The institution specific information (e.g.: how bad is it if the font size has changed but the document is still readable) can be introduced in the form of weigh factors, when compiling a preservation plan that states when to deploy which tools, depending on the institution-specific characteristics.

However, as explained above, this last step is not part of the Testbed application. Testbed results only have to do with the performance of tools on different types of objects, but leave out any institution specific factors.

3 Designing and evaluating Testbed experiments

3.1 Introduction

As stated in the above chapter, the Testbed experiments will

- For migration: test if content, context, appearance, structure and behaviour of digital objects are *preserved* correctly by the particular migration tool.
- For characterisation experiments: test if the characterisation tool manages to *characterise* the appropriate elements of content, context, appearance, structure and behaviour of digital objects correctly.
- For emulation experiments: test if the emulation tool manages to *render* the content, context, appearance, structure and behaviour of digital objects correctly in the new environment / platform.

In the following, for each of these experiment types, the process of selecting evaluation criteria and assessing the outcome of a Testbed experiment is described.

3.2 Characterisation experiment

3.2.1 Designing the experiment

Imagine, for example, a Digital Object type Text, with properties:

- body font family
- body font size
- body font style
- number of words

The user starts defining an experiment, and indicates that

- The experiment is a characterisation experiment with tool X (= tests a characterisation tool X)
- The data is of type Text.

The interface then presents the appropriate, possible evaluation criteria, as follows:

Select	*Correct characterisation of ….* Or more precise: *Similarity of characterised value and reference value of…*
☐	Body font family
☐	Body font size
☐	Body font style
☐	Number of words

The user selects e.g. three criteria: “Similarity of characterised value and reference value of Body font family”, “Similarity of characterised value and reference value of Body font size” and “Similarity of characterised value and reference value of number of words”.

3.2.2 Evaluating the characterisation experiment

After having filled in the other necessary information, the experiment is submitted and run. The result of a characterisation experiment is a file that contains the characterisation of the original data.

The next step is assigning values to the selected evaluation criteria, indicating how successful the experiment was. In this example, we need to assess

- How well tool X has been able to characterise the Body font family
- How well tool X has been able to characterise the Body font size

- How well tool X has been able to characterise the Number of words

To assess the outcome of the experiment and to assign a value to these criteria, we need to know the characteristics / characterisation of the original data and compare them with the characterisation that tool X has given as a result. There are several ways of knowing the characteristics of the original file:

- Use a file from the corpus, i.e. a well-defined object where is it known e.g. that the font size is 12 pt. because the author intentionally chose this font size when compiling the document.
- Visual inspection: e.g. open the original file in a Word processor and conclude that the font size is 12 pt.
- Use an "approved" / "trustworthy" characterisation tool to extract and describe the characteristics.

For example, the characteristics of the original file could be:

- body font size = 12 pt
- body font family = arial narrow

The result of the characterisation tool could be, for instance:

- body font size = 12 pt
- body font family = arial
- <No value found for number of words>

Now, we need to assess this outcome, in other words, define how similar the properties as characterised by the characterisation tool are to the "real" properties of the digital object:

- How similar is the characterised value (12 pt) to the Body font size of the input file (which was 12 pt)?
- Is font family Arial a good characterisation of the Body font family of the input file (which was Arial narrow)? Or: is font family Arial the same as the Body font family of the input file?
- Tool X did not find the value of the number of words in the input file. How do we assess that outcome?

N.B. In evaluating the experiments, we will want to know if the values that the characterisation tool has come up with, are similar to the "actual" values. We do not want to give an opinion whether a specific characterisation is "good enough" or not, because that will vary depending of the circumstances, content holder, usage etc. In other words, we do not weigh the outcomes but try to get results that are as "objective" as possible.

The answers to these questions will be something like, respectively:

- The characterisation by the tool is exactly the same as the "actual" value of the digital object.
- Quite similar, but not completely the same.
- Obviously, that is not a good characterisation: completely different.

Two issues remain:

1. We will need some sort of range of values, or scale, for these outcomes, that indicates the level of similarity of the value as characterised by the tool and the known value of the property of the input file. Possibilities include
    - Give a number from 1, 2, 3.. 10 where 1 is completely different and 10 is exactly the same.
    - Give a number from 1, 2, 3.. 5 where 5 is completely different and 1 is exactly the same.
    - Use a percentage, e.g. 0% is completely different, 100% is exactly the same.
    - Use a range of value, like "completely different", "different", "quite similar" and "exactly the same".

- Use ++, +/-, --.
- Use only two options: “values were the same” and “values were not the same”, or “yes” and “no” or “1” and “0”.

Considerations are:

- More values make it harder to choose the (objectively) correct value.
- An even number of choices is generally preferred to an odd number of options, because people easily choose the middle option. That way, we often get lots of “I don’t know” or “moderate” values, while in most of the times, people do feel that an outcome is a little more “good” or “bad” and hardly ever exactly in the middle.

2. We need a mechanism for assigning the correct value. In general, the two options here are

   - Automated evaluation
   - Manual evaluation

The advantage of automated evaluation is, that it is easier for the user, cost-effective, and more objective. The problems or challenges here, however, are

- How to exactly define a routine or procedure for assigning the correct value, to express how similar two values of a property are. For example, we would need to know that if we measure “arial” instead of “arial narrow”, we will assign an 8 (out of a possible 10) to the characterisation.
- To enable automated evaluation, we will need to describe not only the property values as extracted by the characterisation tool, but also the property values of the original object in a structured way. In other words, the characteristics of the original file and the characterisation by the characterisation tool should use the same language, or, if not, there should be a mapping between the two languages.

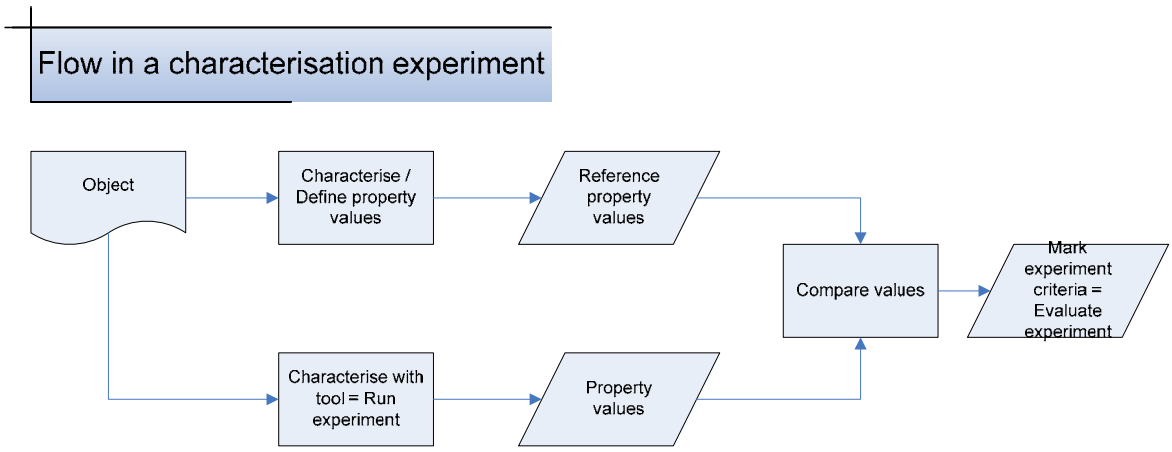
3.2.3 Summary

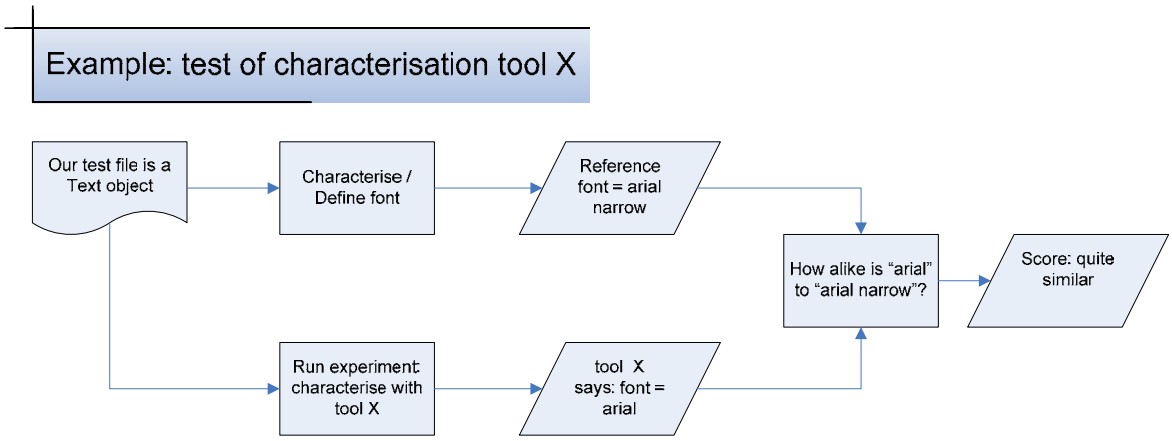
The steps in a characterisation experiment are as follows:

	Step	Example
1.	Select the properties of the Digital Object to focus on.	Digital object is a text; focus on Font family
2.	Define the property values for the digital object. • Use corpora = well-defined object, i.e. object where all the values are known because they have been put there on purpose. • Visual inspection: just look at what you see and write it down. • By use of an “approved” / “trustworthy” characterisation tool.	Font family of a text object is Arial Narrow
3.	Perform characterisation (= run experiment)	
4.	Look at the value the Characterisation tool has produced for Font family	Tool says: font family = Arial
5.	Compare this value to reference value • automatically: possible if the “distance” between Arial Narrow and Arial has been defined.	Decide how similar Arial is to Arial Narrow: similar, but not the same.

	• manually	
6.	Mark this outcome.	Let's mark this as "quite similar".

3.2.4 Diagrams





3.3 Migration experiment

3.3.1 Designing the experiment

Now, imagine a migration experiment on data of type Text. The possible evaluation criteria will then be:

Select	*Correct preservation of ….* Or more precise: *Similarity of value of … in original and migrated object*
☐	Body font family
☐	Body font size
☐	Body font style
☐	Number of words

The user will again select three criteria: “Similarity of value of Body font family in original and migrated object”, “Similarity of value of Body font size in original and migrated object” and “Similarity of value of Number of words in original and migrated object”.

3.3.2 Evaluating the migration experiment

After having filled in the other necessary information, the migration experiment is submitted and run. The result of a migration experiment is a file that contains the same information / “content” as the original data, but in another file format. Therefore, if the original file contains a digital object of type “text”, the migrated file will also contain a digital object of type “text” that will have the same properties as the original digital object.

To assign values to the selected evaluation criteria, indicating how successful the experiment was, we will need to judge how well the values of these properties have been preserved, by migrating the object.

In this example, we need to assess

- How well tool X has been able to preserve the value of the Body font family
- How well tool X has been able to preserve the value of the Body font size
- How well tool X has been able to preserve the value of the Number of words

To mark the outcome of the experiment, we need to know the values of the properties of the original object and compare them with the corresponding property values of the resulting object.

Finding the property values or characteristics of the original object can be done in various ways, as described above:

- Using corpora with known characteristics;
- Using visual inspection of the original file;
- Using an “approved” / “trustworthy” characterisation tool to define the characteristics.

To find the property values or characteristics of the result file, we can use the latter of these three, i.e.

- Using visual inspection of the result (migrated) file;
- Using an “approved” / “trustworthy” characterisation tool to define the characteristics of the result file.

We then need to compare the property values of the original file to the values of the result (i.e. migrated) file and mark how similar they are. The same issues arise as in the characterisation experiment.

3.3.3 Summary

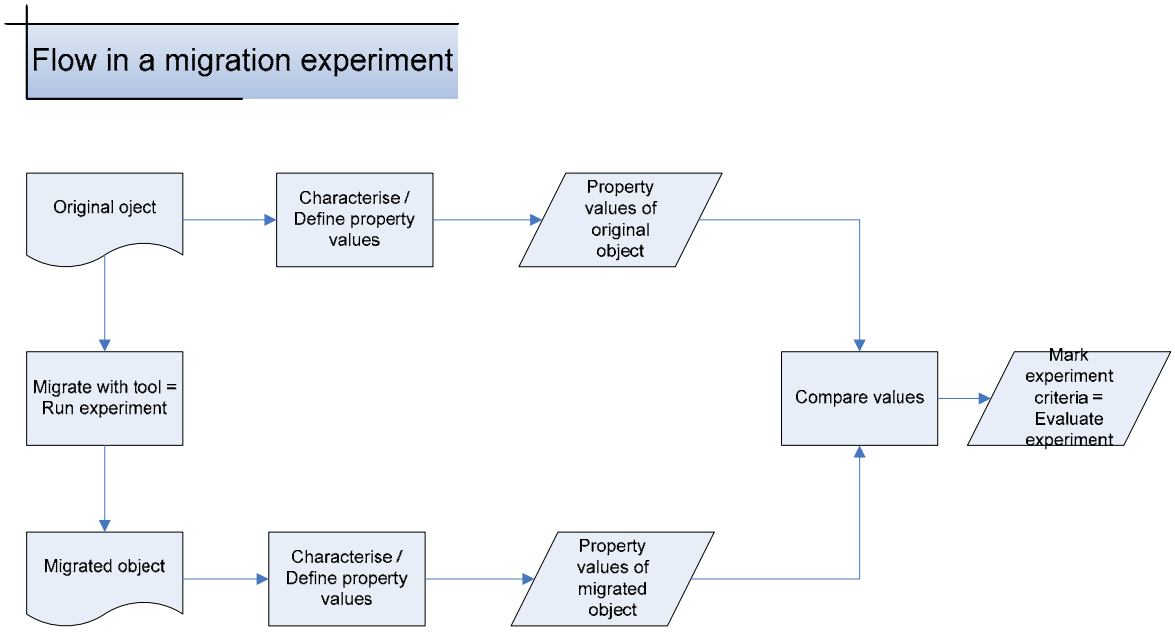
The steps in a migration experiment are as follows:

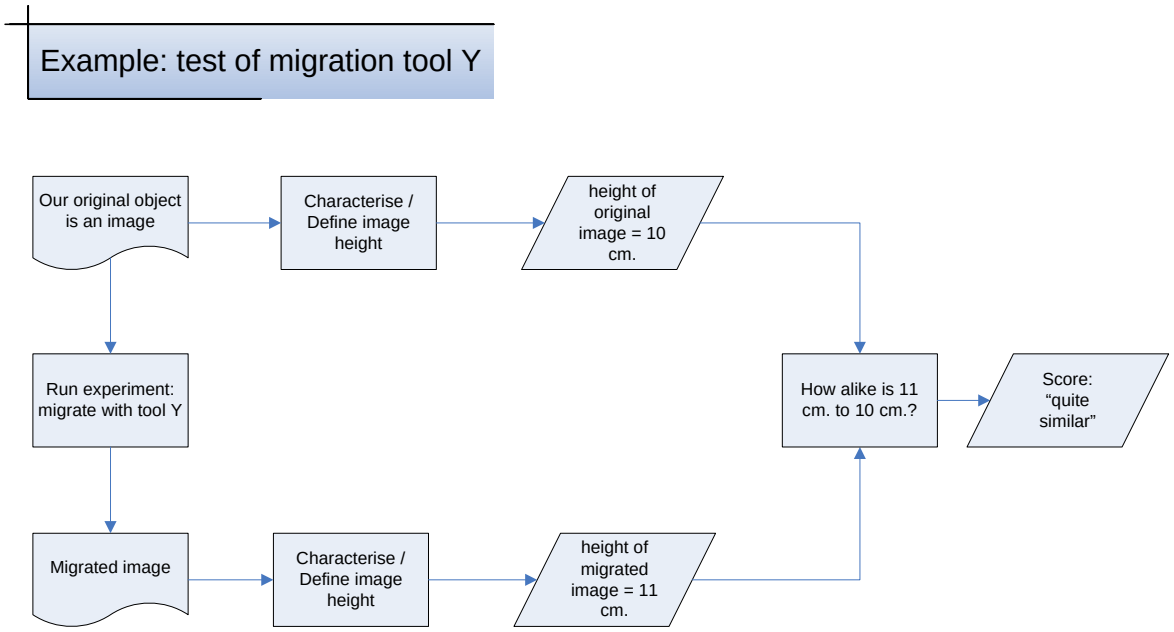
	Step	Example
1.	Select the properties of the Digital Object to focus on.	Digital object is an image; focus on Image height
2.	Define property values for original object: • Use an object from a corpus = a well-defined object, i.e. an object where all the values are known because they have been put there on purpose. • Visual inspection: just look at what you see and write it down. • Use characterisation tool to define these values. N.B. Since in (the first version of) the Testbed, an experiment will consist of one action, e.g. migrate an object from one file format to another, this step is not part of the Testbed experiment, in other	Height of original image = 10cm.

words, this should be done before starting the Testbed experiment.		
3.	Define property values for transformed object: • Visual inspection: just look at what you see and write it down. • Use characterisation tool to define these values.	Height of transformed image = 12cm.
7.	Compare this value to value of original object. • automatically: possible if the “distance” between 10cm. and 12cm. has been defined. • manually	Decide how similar 12cm. is to 10cm.
8.	Mark this outcome.	Let’s mark this as “quite similar”.

See images.

3.3.4 Diagrams





3.4 Emulation experiment

An emulator duplicates (provides an emulation of) the functions of one system with a different system, so that the second system behaves like (and appears to be) the first system. In the digital preservation context, this is a technique that makes it possible to use "old" software on a new platform, in order to be able to see files in their original environment, by using the software that was used to create the object.

Example:

Emulation of a DOS-platform on a new Windows-Vista -machine, in order to use WordPerfect 3.1 on the new machine, in order to see old WordPerfect files in their original context.

Note: the object that is emulated is, in the above example, *not* the WordPerfect file, but the platform needed to use the software to show WordPerfect. However, in this example, we would be interested in the correct rendering of the properties of the WordPefect file.

An emulation experiment is different from characterisation or migration experiment, because is not necessarily something that can be run automatically and produces an output file that can be inspected. Take, for instance, an emulation experiment where one wants to play a DOS-game in its original environment. There, the evaluation criteria will have to do with assessing whether the look and feel of the game in its emulated environment is the same as in the original environment.

3.4.1 Designing the experiment

Let us imagine an experiment where a DOS-game is played in an emulated DOS-environment. Again, it is the environment that is emulated, not the game itself, but we are interested in the correct rendering of the properties of the game. The Digital Object of type Game could have the properties:

- speed
- readability of text
- image resolution

Possible evaluation criteria for an emulation experiment on a digital object of type Game could thus be:

Select	*Correct rendering of….* Or more precise: *Similarity of value of … of object in original and emulated environment*
☐	Speed

☐	Readability of text
☐	Image resolution

Let us say the user selects all three criteria: “Correct rendering of speed”, “Correct rendering of readability of text” and “Correct rendering of Image resolution”.

3.4.2 Evaluating the emulation experiment

The exact way of running the emulator in the Testbed is still to be defined, but at some point the user will be able to play the game in the emulated old environment. To assign values to the selected evaluation criteria, indicating how successful the experiment was, we will need to judge how well the values of the properties of the game are rendered in the emulated environment, compared to the property values in the original environment.

In this example, we need to assess

- How well emulation tool X has been able to render the original Speed
- How well emulation tool X has been able to render the original Readability of text
- How well emulation tool X has been able to render the original Image resolution

To mark the outcome of the experiment, we need to know the values of the properties of the object in its original environment and compare them with the corresponding property values of the object in the emulated environment.

Finding the property values of the object in its original context can again be done in various ways.

If the object that is considered is e.g. a text document, options are:

- Using corpora with known characteristics;
- Using visual inspection of the original file;
- Using an “approved” / “trustworthy” characterisation tool to define the characteristics.

In case of using any software tools to define the property values, it must be kept in mind that the software must be able to run on an “old” platform, namely the platform that is to be emulated, e.g. DOS.

If the object that is considered is e.g. a game, it is more likely that the values are defined manually.

To find the property values of object in the emulated environment, the result file, we can again use the latter of these three, i.e.

- Using visual inspection of the object.
- Use of an “approved” / “trustworthy” characterisation tool to define the characteristics of the resulting object.

Again, in case of using any software tools to define the property values, this software must be able to run on an “old” platform, namely the platform that is being emulated, e.g. DOS.

We then need to compare the property values of the objects in their original environment to the values of the object in the emulated environment and mark how similar they are.

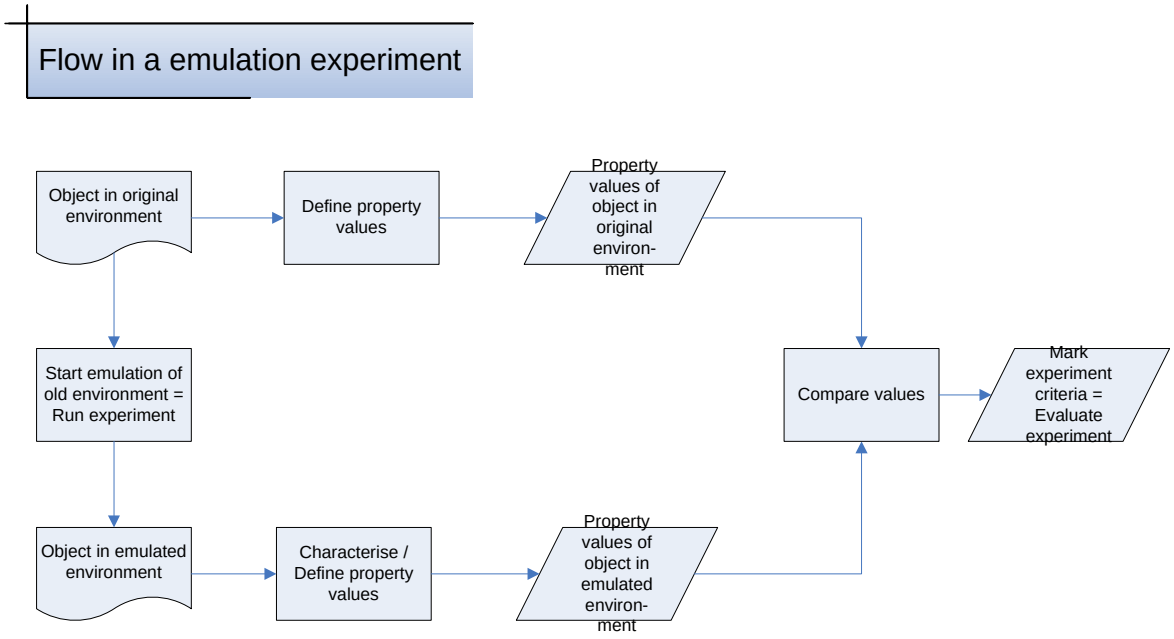
3.4.3 Summary

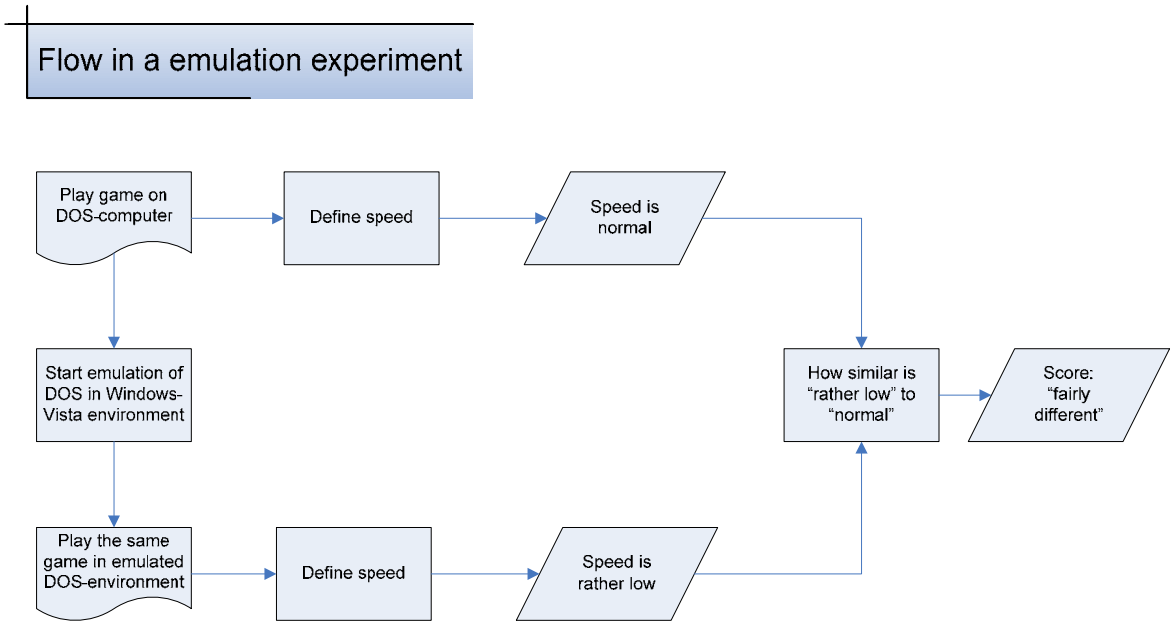
The steps in an emulation experiment are as follows:

	Step	Example
1.	Select the properties of the Digital Object to focus on.	Digital object is a game; focus speed.
2.	Define property values for object in original context: • Use an object from a corpus = a well-defined object, i.e. an object where all the values are known because they have been put there on purpose. • Visual inspection: just look at what you see and write	We play the game in the original DOS-environment and experience that the speed is normal.

	it down. • Automatically: use characterisation tool to define these values.	
3.	Define property values for object in emulated environment: • Visual inspection: just look at what you see and write it down. • Use characterisation tool to define these values.	We now play the game in the DOS-environment emulated on a Windows-Vista-machine and experience that the game is rather slow, we have to wait after every keystroke.
9.	Compare this value to value of original object. • automatically: not possible in this example because we have not used a formal range of values to mark the two speeds. • manually	Decide how similar “normal speed” is to “rather slow”.
10.	Mark this outcome.	Let’s mark this as “fairly bad”.

3.4.4 Diagrams




Flow in a emulation experiment
Play game on DOS-computer
Define speed
Speed is normal
Start emulation of DOS in Windows-Vista environment
How similar is "rather low" to "normal"
Score: "fairly different"
Play the same game in emulated DOS-environment
Define speed
Speed is rather low

4 Suggestions for incremental development of the Testbed

Ideally, experiments should be evaluated as automated as possible, with the least possible user interaction, to ensure the most objective results. However, this would require the availability of well-described reference material for every type of experiment, and a way to compare these references to the experiment outcomes automatically. Therefore, the following suggestion for incremental development of the Testbed.

4.1 Actions in and outside the Testbed application

- There will be experiments which will consist of one action, e.g. characterising of an object. When, considering such a characterisation experiment, compiling a reference characterisation of the object is, strictly speaking, not part of the Testbed experiment, i.e. is not done via the Testbed user interface. In other words, the reference characterisation of the object should be compiled before starting the Testbed experiment.
- Likewise, in case of a simple migration experiment that consists of only one step, namely migrating an object from one file format to another, neither defining the properties (characterising) of the original object, nor defining the properties (characterising) of the migrated object, are parts of the Testbed experiment, i.e. are done via the Testbed user interface.
- When carrying out a workflow that consists of more then one step, the sequence of actions (e.g. in a migration experiment: 1. characterising the original object, 2. migrating the original object, 3. characterising the migrated object) can all be carried out in the Testbed.

4.2 Evaluation criteria

In order to evaluate experiments objectively, we need a mechanism to compare the property values of an object to the corresponding reference value:

- For a characterisation experiment: compare to property in e.g. corpora;
- For migration experiment: compare to property in original (not migrated) object;
- For emulation experiment: compare to property of same object in original environment.

Really objective – and possibly automated – evaluation requires:

- A way to “measure” the property value: a unit, or metric.
- An algorithm to calculate the “distance” between two values.

For some properties, assigning a value to the property is relatively easy, e.g. font size can be expressed in pt. The similarity between e.g. 12pt. and 10pt. could be calculated as 12/10, or 10/12, or expressed in percentages, a scale from 1 to 5, etc.

For other properties, assigning a property value is not difficult, because the possible values are known, but comparing two value is harder. E.g. how similar are “arial” and “arial narrow”? Or “arial” and “universe”?

For a third group of properties, assigning a property value can also be hard, or at least finding a scale to express the possible values is difficult. How, for instance, do you describe “look and feel”?

For this group of properties, comparing two values and assessing the level of similarity is difficult.

Therefore, for the first version of the Testbed, let us use a simple set of evaluation criteria to express the similarity of two values, like:

- Exactly the same

- Quite similar
- Quite different
- Completely different

We could discuss about more scales and metrics later, to facilitate more automated and objective evaluation in later versions.

4.3 Automatic evaluation

In the first version of the Testbed, evaluating the experiment, i.e. comparing reference values of properties to outcomes of the experiment, will be done manually, or, at least not via the Testbed user interface. (To be precise, it may be that the reference characterisation and the resulting characterisation are indeed compared automatically, but that this comparison is actually done “outside” the Testbed application. In any case, the resulting evaluation marks will be filled in by the user manually.)

5 Use of Testbed Results

5.1 Introduction

The following paragraphs describe how the Testbed results could be used in the registry of tools, and in compiling a preservation plan.

5.2 Adding information to the tools registry

After some experiments, we would have Testbed results like, for example:

Characterisation tool X, Digital Object Type: Image	***Characterisation of Width***	***Characterisation of Height***	***Characterisation of Colour depth***	***Characterisation of Resolution***
Experiment 1	quite similar	quite similar	quite similar	exactly the same
Experiment 2			quite similar	quite similar
Experiment 3	exactly the same	exactly the same	completely different	exactly the same
Experiment 4	quite similar	quite similar		
Experiment 5	exactly the same	exactly the same	exactly the same	exactly the same

and

Migration tool Y, Digital Object Type: Text	***Migration of Font type***	***Migration of Font Size***	***Migration of Headers***	***…***
Experiment 1	quite similar	quite similar		
Experiment 2	exactly the same	exactly the same	exactly the same	exactly the same
Experiment 3			completely different	completely different
Experiment 4	quite similar	quite similar		

We could think of an algorithm to aggregate the values per experiment to an average value for all the experiments:

Characterisation tool X, Digital Object Type: Image	***Characterisation of Width***	***Characterisation of Height***	***Characterisation of Colour depth***	***Characterisation of Resolution***
Experiment 1	quite similar	quite similar	quite similar	exactly the same
Experiment 2			quite similar	quite similar
Experiment 3	exactly the same	exactly the same	completely different	exactly the same
Experiment 4	quite similar	quite similar		
Experiment 5	exactly the same	exactly the same	exactly the same	exactly the same
Average	**quite similar**	**quite similar**	**quite similar**	**exactly the same**

and

Migration tool Y, Digital Object Type: Text	***Migration of Font type***	***Migration of Font Size***	***Migration of Headers***	***Migration of Page numbering***
Experiment 1	quite similar	quite similar		
Experiment 2	exactly the same	exactly the same	exactly the same	completely different

Experiment 3			quite different	completely different
Experiment 4	quite similar	quite similar		
Average	**quite similar**	**quite similar**	**quite similar**	**completely different**

Many variations for an aggregation algorithm are possible, e.g. leaving out the best and the worst value. Our goal would be to get information on how a tool performs on each of the possible properties of a digital object type:

Characterisation tool X, Digital Object Type: Image	***Average value for characterisation of…***
Width	quite similar
Height	quite similar
Colour depth	quite similar
Resolution	exactly the same

and

Migration tool Y, Digital Object Type: Text	***Average value for migration of …***
Font type	quite similar
Font size	quite similar
Headers	quite similar
Page numbering	completely different

We could even think of a further aggregation over all the values per property. Again, many ways to do this are possible. This would give us an average value for each tool, per digital object type. It would be a good idea to keep track of at least the number of experiment, the average value it is based on, and maybe the best value and the worst value, like this:

Characterisation tool X	***Average value for characterisation of Digital Object Type…***
Text	quite similar
Image	quite similar
…	..

Migration tool Y	***Average value for migration of Digital Object Type…***
Text	quite similar
Image	quite similar
…	..

It might be wise, especially for migration tools, to keep track of the file formats that are used in the migration. That way, we could distinguish between e.g. migration of Text files from .doc to .odf, migration of Text files from .txt to .xml etc.

The resulting values could be written back to the tools registry regularly, and be of use in compiling a preservation plan without having to consult the Testbed results database.

5.3 Using the information for the compilation of a preservation plan.

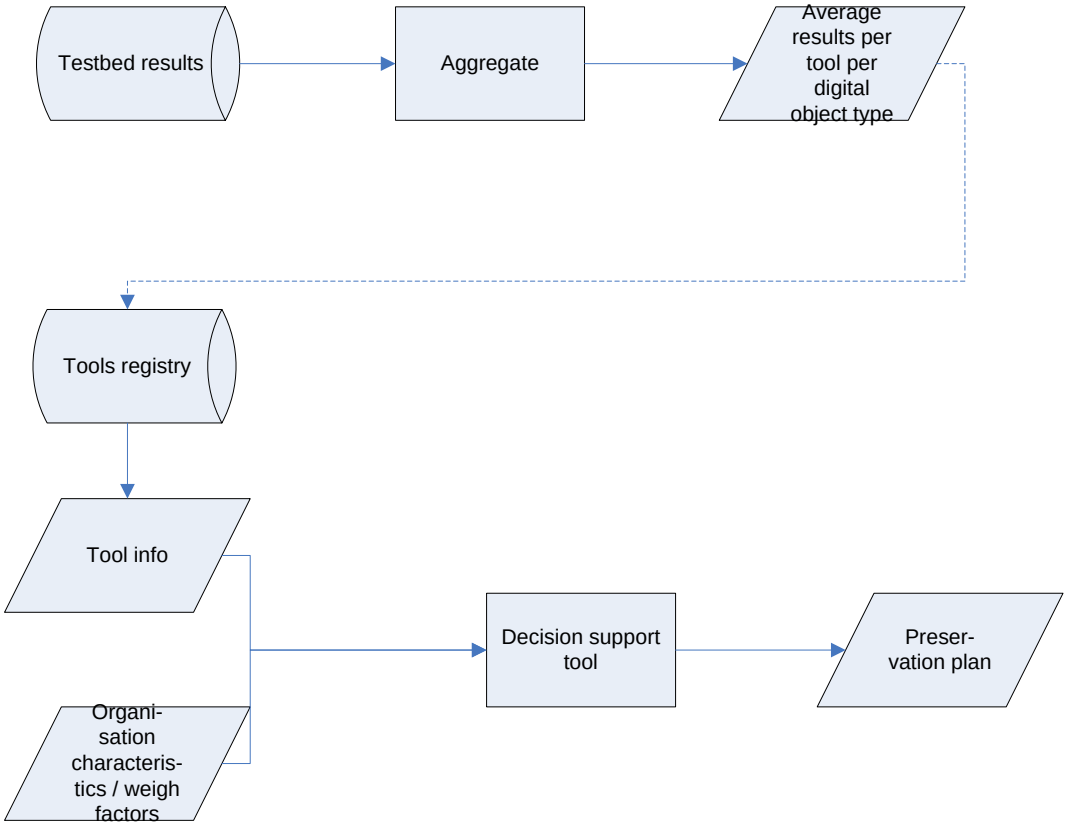
As stated before, the Testbed results are as objective as possible and do not take into account any organisation specific elements, like user profile, collection profile etc.

To choose which tools are best for use in a specific organisation, of course these topics have to be taken into account. For instance, the fact that a certain migration tool is very quick and cheep but, when used for transforming audio files to another format, makes the highest notes inaudible might

not matter for an institution A, the collection of which contains only a few audio files with mainly speech, but might be unacceptable for an institution B, the collection of which consists mainly of high quality audio-visual material.

When compiling a preservation plan, institution A will assign a lower weigh factor to the correct preservation of high notes than institution B. A decision supporting instrument that will select the best tools, given the assigned weigh factors for each decision element. Such an instrument could read the tools registry, looking for all tools suitable for migrating audio files and reading the average values that have been assigned to them, by aggregation of the Testbed results.

For institution A it could suggest an audio migration tool with a value of at least “quite similar” for the preservation of high notes. For institution B it would suggest only audio migration tools with a value “exactly the same”.
See image.



Appendix A

A.1 Introduction

A final list of Digital Object Types to use in the Testbed has still to be decided on, probably in close collaboration with sub-projects PC, PP and TB.

A provisional list of Digital Object Types is given below, as is a provisional list of the associated properties that will be used as evaluation criteria in the following way:

- For characterisation experiments, the possible criteria will be: “Correct characterisation of <digital object property>
- For migration experiments, the possible criteria will be: “Correct migration of <digital object property>
- For emulation experiments, the possible criteria will be: “Correct rendering of <digital object property>

A.2 Provisional list of Digital Object Types

Name	Description
Text	Typical file formats include: .doc, .pdf, .txt…
Image	Typical file formats include: .tiff, .bmp, .jpg, .gif, …
Audio	Typical file formats include: .wav, .mp3, …
Video	
Application	Typical file formats include: .exe
Database	
Website	
Mixed	In case of Test Objects with mixed content, the properties of the selected test objects should be listed, e.g. Text + Image selected will select all properties of Text and all properties of Image.

There are many ways a list of this sort can be compiled and this could lead to all sorts of scientific discussion. If we could come to an exhaustive list at this moment, it would be great, but it is not really necessary from a Planets / Testbed point of view.

However, from a Planets / Testbed point of view, only digital object types have to be included in the list for which partners intend to execute experiments. If, for example, no-one is concerned about the migration of emails and there are no experiments planned to carry out on email, there is no need for a discussion whether or not email should be considered a distinctive digital object type.

Ideally, we could come to an exhaustive list at this moment, but if not, for every object that is added to the list, a corresponding list of properties should be provided.

A.3 Provisional list of Properties per Digital Object Type

A.3.1 Text

Name	Definition	Description	Scale

Number of levels	Defines the number of subdivision levels beneath the highest level in a text.	If the text is e.g. divided in parts, which each have chapters, which are divided in paragraphs, the number of levels would be 3. For most emails, the number of levels will be 0 if they are not subdivided. Makes it possible to decide whether the structure has been preserved / characterised. "structure"	1..n
Number of level 1 subdivisions	Identifies the number of highest level parts the text is divided in.	If the text has a table of contents, a preface and 3 chapters, this number could be 5. Makes it possible to decide whether the structure has been preserved / characterised, and e.g. no chapters have been lost. "structure"	1..n
Number of footnotes	Indicates the number of footnotes in the text.	Count the number of footnotes in the entire text. "structure"	1..n
Number of footnote references (indicators).	Indicates the number of footnote references (indicators).	Count the number of footnote references (indicators) in the entire text. "structure"	1..n
Use of headers	Indicates whether or not headers are used in the text.	Yes if (one or more) headers are used, no if not. "structure"	yes/no
Number of different headers	Indicates the number of different headers used in the text.	Count the number of different headers, e.g. each chapter could have a different header. "structure"	1..n
Use of footers	Indicates whether or not footers are used in the text.	Yes if (one or more) footers are used, no if not. "structure"	yes/no
Number of different footers	Indicates the number of different footers used in the text.	Count the number of different footers e.g. each chapter could have a different footer. "structure"	1..n
Number of tables	Indicates the number of tables used in the text.	Count the number of tables. "structure"	1..n
Number of table captions	Indicates the number of table captions.	Count the number of table captions. "structure"	1..n
Number of images	Indicates the number of image captions used in the text.	Count the number of image captions. "structure"	1..n
Number of image captions	Indicates the number of images used in the text.	Count the number of images. "structure"	1..n
Number of graphics	Indicates the number of graphics used in the text.	Count the number of graphics. "structure"	1..n

Number of graphics captions	Indicates the number of graphic captions used in the text.	Count the number of graphic captions. "structure"	1..n
Number of formulas	Indicates the number of formulas used in the text.	Count the number of formulas. "structure"	1..n
Use of signatures	Indicates whether or not the text contains signatures.	Yes if text contains one ore more signatures, no otherwise. "structure"	yes/no
Use of signatures	Indicates whether or not the text contains any signatures.	Yes if text contains one ore more signatures, no otherwise. Can be in any form, e.g. digital signature ir watermark-like. "structure"	yes/no
Use of watermarks / background images	Indicates whether or not the text contains a watermark or background image.	Yes if text contains (one ore more) watermarks or background images. "structure"	yes/no
Number of working hyperlinks	Indicates the number of working hyperlinks used in the text.	Count the number of working hyperlinks. "behaviour"	1..n
Number of pages	Defines the number of pages.	Count the number of pages, including the front page. "appearance"	1..n
Number of lines	Defines the number of lines.	Count the number of lines. "appearance"	1..n
Number of characters	Defines the number of characters.	Count the number of characters. "appearance"	1..n
Page margin-top	Unused space at the top of the document above the text.	"appearance"	cm. / inch.
Page margin-right	Unused space at the right of the document above the text.	"appearance"	cm. / inch.
Page margin-bottom	Unused space at the bottom of the document above the text.	"appearance"	cm. / inch.
Page margin-left	Unused space at the left of the document above the text.	"appearance"	cm. / inch.
Page height	Total height of the page.	"appearance"	cm. / inch.
Page width	Total width of the page.	"appearance"	cm. / inch.
Number of columns	Identifies the number of columns.	Count the number of columns on a page. No columns means the number	1..n

		is 1. If there are several instances of "plain text" with different values, choose one to focus on. (for first version) "appearance"	
Columns spacing	Identifies the white space between two columns.	Define the unused space between two columns. If there are several values for this property, choose one to focus on. (for first version) "appearance"	cm. / inch.
Letter / word spacing	Identifies the letter / word spacing of plain text.	Indicate if the white space of a line is evenly distributed between the letters or words. If there are several values for this property, choose one to focus on. (for first version) "appearance"	proportional / absolute
Paragraph font size	Font size of plain text.	The font size of plain text. If there are several instances of "plain text" with different values, choose one to focus on. (for first version) "appearance"	px. / pt/ / cm. / inch
Paragraph font style	Font style of plain text.	The font style of plain text. If there are several instances of "plain text" with different values, choose one to focus on. (for first version) "appearance"	category
Paragraph font colour	Identifies the font colour of plain text.	The font colour of plain text. If there are several instances of "plain text" with different values, choose one to focus on. (for first version) "appearance"	string
Paragraph line spacing	Identifies the line spacing of plain text.	The spacing between two lines of plain text. If there are several instances of "plain text" with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Paragraph spacing-top	Identifies the space from top of paragraph to element above.	Space from top of paragraph to element above. If there are several instances of "plain text" with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Paragraph spacing-right	Identifies the space from right of paragraph to element at the right.	Space from right of paragraph to element at the right. If there are several instances of "plain text" with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Paragraph spacing-bottom	Identifies the space from bottom of paragraph to element below.	Space from right of paragraph to element below. If there are several instances of "plain text" with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Paragraph	Identifies the space from left	Space from left of paragraph to	cm. /

spacing-left	of paragraph to element at the left.	element at the left. If there are several instances of "plain text" with different values, choose one to focus on. (for first version) "appearance"	inch.
Paragraph indentation	Identifies the space from page margin to left of first line of paragraph.	Space from the page margin to the left of the first line of the paragraph. If there are several instances of "plain text" with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Header font size	Font size of headers.	The font size of headers. If there are several instances of headers with different values, choose one to focus on. (for first version) "appearance"	px. / pt/ / cm. / inch
Header font style	Font style of headers.	The font style of headers. If there are several instances of headers with different values, choose one to focus on. (for first version) "appearance"	category
Header font colour	Identifies the font colour of headers.	The font colour of headers. If there are several instances of headers with different values, choose one to focus on. (for first version) "appearance"	string
Header line spacing	Identifies the line spacing of headers.	The spacing between two lines of headers. If there are several instances of headers with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Header spacing-top	Identifies the space from top of header to element above.	Space from top of header to element above. If there are several instances of headers with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Header spacing-right	Identifies the space from right of header to element at the right.	Space from right of header to element at the right. If there are several instances of headers with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Header spacing-bottom	Identifies the space from bottom of header to element below.	Space from right of header to element below. If there are several instances of headers with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Header spacing-left	Identifies the space from left of Header to element at the left.	Space from left of header to element at the left. If there are several instances of headers with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.

Footer font size	Font size of footers.	The font size of footers. If there are several instances of footers with different values, choose one to focus on. (for first version) "appearance"	px. / pt/ / cm. / inch
Footer font style	Font style of footers.	The font style of footers. If there are several instances of footers with different values, choose one to focus on. (for first version) "appearance"	category
Footer font colour	Identifies the font colour of footers.	The font colour of footers. If there are several instances of footers with different values, choose one to focus on. (for first version) "appearance"	string
Footer line spacing	Identifies the line spacing of footers.	The spacing between two lines of footers. If there are several instances of footers with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Footer spacing-top	Identifies the space from top of footer to element above.	Space from top of footer to element above. If there are several instances of footers with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Footer spacing-right	Identifies the space from right of footer to element at the right.	Space from right of footer to element at the right. If there are several instances of footers with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Footer spacing-bottom	Identifies the space from bottom of footer to element below.	Space from right of footer to element below. If there are several instances of footers with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Footer spacing-left	Identifies the space from left of footer to element at the left.	Space from left of footer to element at the left. If there are several instances of footers with different values, choose one to focus on. (for first version) "appearance"	cm. / inch.
Human readability	Indicates how well the text is readable.	Grade how well the text is readable. 1 is completely different, 10 is excellent.	1..10
Machine readability	Indicates which percentage of the characters of the text are machine readable.	Define this value with a reference application, e.g. an OCR-application. 0% means nothing readable, 100% means every character readable.	0..100

A.3.2 Image

Name	Definition	Description	Scale

Image height	Identifies the height of the image.	Size from top to bottom. "appearance"	cm. / inch
Image width	Identifies the width of the image.	Size from left to right. "appearance"	cm. / inch
Color depth	Describes the number of bits used to represent the colour of a single pixel	Can not be determined by visual inspection (you cannot simply see it) but only by a tool.	bits/pixel
Colour space defined	Indicates whether or not the colour space has been defined.	The colour space defines how an application interprets the value of a colour. Can not be determined by visual inspection (you cannot simply see it) but only by a tool.	yes / no
Size of smallest detail	Defines the smallest detail that can be distinguished in an image.	Can be resolution in a .jpeg or .bmp image, but different for a vector image.	cm. / inch

A.3.3 Audio[1]

Name	Definition	Description	Scale
Audio resolution	Describes the band-width	Describes the band-width	Bit per sample
Tracking	p.m.	p.m.	p.m.
Level	p.m.	p.m.	p.m.
Dynamic	Defines the difference between the loudest and the softest sounds.	p.m.	p.m.
Spectral components	??	p.m.	yes / no
Contains phasing	??	p.m.	yes / no
Signal-to-noise-ratio	Compares the level of a desired signal (such as music) to the level of background noise. The higher the ratio, the less obtrusive the background noise is.	p.m.	0..1
Drop-Out	Short periods during which the sound is interrupted	Count or estimate of the average number of drop-outs per minute in the audio record.	1..n

[1] The list below is based partly on C. Rauch e.a., *Preservation. A Framework for Documenting the behaviour and Functionality of Digital Objects and Preservation Strategies* (Draft DELOS Report d.d. 11 April 2005)

Sample Rate	Describes how often per second the data source is touched.	Cannot be defined manually, tool needed.	1..n
Water marking	Describes if the object contains a watermark	Cannot de defined manually, tool needed.	yes / no
Multitrack	Describes the number of channels used in recording the sound.	Give the number of tracks. Cannot be heard, should be known in advance.	1..n
Speed variance	Describes whether the speed gets sometimes a little faster or a little slower.	p.m.	yes / no
Mono/Stereo	Describes whether the recording is in mono or in stereo	p.m.	Mono / Stereo
Duration	Length of recording	Length of recording, induding "silence" before and after sound.	Seconds

A.3.4 Video[2]

Name	Definition	Description	Scale
Resolution	The granularity of the picture	p.m.	1..n
Picture Drop-Out	Small errors in the picture stream, e.g. a short black screen.	Count or estimate of the avarage number of drop-outs per minute in the video record.	1..n
Sound Drop-Out	Small errors in the sound, such as no sound for a short term	Count or estimate of the avarage number of drop-outs per minute in the video record.	1..n
Picture Aspect Ratio	Relation of width to length of a picture	p.m.	0..n
High Definition	Defines if the video conforms to the High Definition standard.	The amount of information transmitted to the screen. High Definition is the upcoming standard for television sets.	yes / no
Frame rate	The speed of change between the pictures	In frames per second	1..n
Stereo	Describes whether files are opened with one or two sound channels	yes for stereo, no for mono.	yes / no
Signal representati	Indicates the colour representation scheme, such	p.m.	string

2 2 The below list is based partly on C. Rauch e.a., *Preservation. A Framework for Documenting the behaviour and Functionality of Digital Objects and Preservation Strategies* (Draft DELOS Report d.d. 11 April 2005)

on	as RGB		
Picture-audio synchronisation	Indicates whether sound and the adjusted pictures are shown at exactly the same time.	p.m.	yes / no