Technische Universität Wien

Vienna University of Technology

MASTERARBEIT

A Functional Model of a Digital Preservation Testbed

ausgeführt am

Institut für Softwaretechnik und Interaktive Systeme

der Technischen Universität Wien

unter der Anleitung von

ao.univ.Prof. Dr. Andreas Rauber

durch

Dipl.-Ing. Christoph Becker
9952542

Mayerhofgasse 7/13
1040 Wien

Wien, im *April 2007*

Abstract

A large part of the cultural and scientific heritage of our time is available only in digital form. The growing heterogeneity and complexity of digital file formats have turned preserving these materials and ensuring proper long-term access to them into a pressing challenge. Digital preservation has thus become an active research area over the last years, and a number of different solutions have been proposed to tackle the issues of preserving digital objects.

These proposed solutions and tools all have very specific characteristics and differ significantly in their strengths and weaknesses. Testing and evaluating preservation strategies is thus a complex matter, partly due to the lack of a ground truth against which solutions can be evaluated. The digital preservation community is clearly missing a kind of benchmark on which to evaluate preservation approaches.

To arrive at well-informed, consistent, comparable, and accountable documentation of the characteristics, strengths and weaknesses of preservation solutions, a controlled environment with benchmark data is needed. Several related efforts deal with these tasks. The DELOS Digital Preservation Cluster is working on criteria for a benchmark corpus; the PLANETS project is building a testbed system on which preservation experiments can be run in a well-defined and consistent manner to arrive at comparable measurements of the special characteristics of preservation tools and services.

This thesis proposes the usage model for such a testbed. We lay out the foundations of digital preservation, describe ongoing efforts to tackle the main challenges, and the issues of testing and evaluating different solutions. We then analyse the requirements on a digital preservation testbed and provide an overview of related efforts.

The main part of this thesis then describes the functional model of a testbed for digital preservation by using the use case methodology. We describe the user profiles and roles that are interacting with the system and the main blocks of functionality that the system needs to provide. We then describe the use cases of the testbed in detail.

Lastly, we point out the current state of development and future steps to be taken.

Kurzfassung

Ein großer Teil des kulturellen und wissenschaftlichen Erbes unserer Zeit ist nur in digitaler Form verfügbar. Die wachsende Heterogenität und Komplexität von digitalen Dateiformaten bewirken, dass die Aufgabe, diese Materialien aufzubewahren und auf zuverlässige Weise langfristig zugänglich zu machen, eine dringende Herausforderung darstellt. Digitale Langzeitarchivierung ist daher in den letzten Jahren zu einem aktiven Forschungsfeld geworden, und eine Anzahl verschiedener Lösungsansätze sind vorgeschlagen worden, um die Schwierigkeiten bei der langfristigen Archivierung von digitalen Objekten zu bewältigen.

Diese Lösungen und Werkzeuge haben alle sehr spezifische Charakteristiken und Besonderheiten und unterscheiden sich signifikant in ihren Stärken und Schwächen. Das Testen und Evaluieren derartiger Lösungen ist daher eine komplexe Aufgabe. Verstärkt werden die Schwierigkeiten durch das Fehlen von Referenzwerten, mit denen die Messungen neuer Lösungen verglichen werden könnten. Die Gemeinschaft der betroffenen Institutionen und Forscher brauchen daher dringend einen Vergleichstest zur Evaluierung von Lösungen und Werkzeugen.

Um konsistente, vergleichbare und verantwortliche Dokumentationen über die Charakteristiken, Stärken und Schwächen von Werkzeugen zu ermöglichen, ist eine kontrollierte Umgebung mit Benchmark-Daten notwendig. Mehrere zusammenhängende Initiativen befassen sich mit dieser Aufgabe.

Der *DELOS Digital Preservation Cluster* arbeitet an Kriterien für einen Testdaten-Korpus; das Projekt *PLANETS* erstellt eine Test-Plattform, auf der Experimente in wohldefinierter und konsistenter Weise ausgeführt werden können, um vergleichbare Messdaten über die speziellen Eigenschaften von Werkzeugen und Diensten zu erhalten.

Diese Arbeit stellt das funktionale Modell für ein derartiges System vor. Wir erläutern die Grundlagen digitaler Langzeitarchivierung, beschreiben aktuelle Initiativen und Lösungansätze und gehen auf die Probleme beim Testen und Bewerten verschiedener Werkzeuge ein. Wir analysieren daraufhin die Anforderungen an eine Test-Plattform zur Evaluierung verschiedener Lösungen und stellen verwandte Ansätze vor.

Der zentrale Teil dieser Arbeit beschreibt das funktionale Modell einer Test-Plattform für digitale Langzeitarchivierung. Wir analysieren Benutzerprofile und Rollen, die mit dem System interagieren, sowie die wesentlichen Blöcke der Funktionalität, die das System bereitstellt, und beschreiben die Anwendungsfälle der Plattform im Detail.

Abschließend legen wir den aktuellen Stand der Entwicklung dar und gehen auf die bevorstehenden nächsten Schritte ein.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Digital objects have become the dominant way that we create, shape, and exchange information. They increasingly contain essential parts of our cultural, intellectual and scientific heritage; they form a central part of our economy, and increasingly shape our private lives.

The ever-growing heterogeneity and complexity of digital file formats together with rapid technological changes turn the preservation of digital information into a pressing challenge. The challenge is to keep electronic data accessible, viewable, and usable for the future, to ensure the survival of our digital artifacts when the original software or hardware to interpret them correctly becomes unavailable [UNE03].

A number of approaches to digital preservation have been proposed over the years, the most prominent ones being migration and emulation. Yet, the decision on which preservation solution best fits a specific purpose within a particular institutional context remains a complex and difficult one. Preservation Planning approaches provide support in reaching well-founded and solidly documented decisions, but they also rely on in-depth comparisons of particular characteristics of approaches, e.g. with regard to the scalability of migration tools converting office documents to PDF format. These comparisons need a well-defined, stable environment to enable the large-scale evaluation of the characteristics of different tools and approaches when applied to large collections and across different institutions.

To achieve this, the PLANETS project[1] is developing and establishing a testbed which will function as a controlled environment to test tools and solutions for digital preservation.

Integrated services will be explored, evaluated and refined in the

[1] http://www.planets-project.eu

> preservation Testbed. The results of ongoing evaluation within the Testbed will provide essential feedback to guide the development of PLANETS tools and services.[2]

The PLANETS Testbed is partly based on the Digital Preservation Testbed developed in the Dutch National Archive[3]. However, it goes far beyond the former with regard to the integration of tools and services. It will allow semi-automated definition and execution of large-scale experiments over a distributed service network.

Institutions will use the testbed for example to

- test and evaluate new tools and services for both preservation action services such as migration and emulation tools or preservation characterisation services,
- compare key characteristics of different tools applied to benchmark content, or
- evaluate tools on their own collections uploaded to the platform.

The testbed provides a stable and consistent platform to enable meaningful comparison of experiment results, a repository for test data, the procedural framework and a usage model for conducting experiments.

Three important aspects thus need to be considered when developing and establishing a controlled environment for testing and evaluating digital preservation tools like the PLANETS Testbed.

1. A system or platform for testing,
2. Benchmark data, and
3. A usage model.

While the DELOS Digital Preservation Cluster[4] is defining criteria for digital preservation benchmark corpora and working towards the creation of these corpora, this thesis focuses on the functional model of the testbed. We will analyse the concepts of the Dutch Testbed, lay out the foundations of the PLANETS architecture and goals and describe the functional model of the new testbed using the use case methodology.

The remainder of this thesis is structured as follows.

[2] `http://www.planets-project.eu/docs/comms/Planets_flyer_2006.pdf`

[3] `http://www.nationaalarchief.nl`

[4] `http://www.dpc.delos.info`

Chapter 2 provides an overview of digital preservation foundations, ongoing activities and initiatives as well as current challenges. It outlines the foundations of evaluating and testing preservation strategies, describes the challenges of creating benchmark corpora for digital preservation, and gives a short introduction on the use case methodology which is employed to describe the functional model of the testbed.

Chapter 3 outlines the principles of the Dutch Testbed, discusses the motivation for building a new testbed system and gives an overview of its principal architecture. We will also put the PLANETS Testbed in context by introducing the PLANETS project and its distributed service architecture.

Chapter 4 describes the functional model of the PLANETS Digital Preservation Testbed. The main tasks that are described by this usage model include

- The definition of experiment settings, including configuration of tools to be tested and the data sets that shall be used for testing,
- A process for experiment submission, approval, and execution; and
- Access to the registries containing experiments, test data, and results.

We will also cover administrative tasks and additional functionality beside these core tasks.

The last chapter summarizes the work of this thesis and points out further work and research directions.

Chapter 2

Related Work

This chapter starts with an introduction to the challenges of digital preservation. It gives an overview of digital preservation strategies that have been devised over the years and major initiatives that have been started all over the world.

We then introduce the issue of testing and evaluating digital preservation strategies. An important requirement for being able to thoroughly evaluate digital preservation strategies in an objectively comparable and repeatable manner is the existence of an open benchmark data corpus. We introduce several related efforts for defining data corpora for benchmarking purposes and describe the specific challenges for creating corpora for digital preservation.

The chapter concludes with a short introduction on the use case methodology that we rely on for specifying the functional model of the testbed.

2.1 Digital Preservation

An increasing amount of cultural heritage material, legal, and scientific information is born-digital or only available in digital form.

Yet, current systems, platforms and environments have been and still are constantly evolving. The ability of accessing and using digital information usually depends on a particular version of a programm on a specific computer platform. The heterogeneity and complexity of these platforms and the employed digital formats turn the preservation of digital objects to ensure long-term access to the contained information into a difficult task.

Digital preservation deals with the long-term storage and access to digital objects. The Digital Preservation Coalition defines it as *the series of managed activities necessary to ensure continued access to digital materials* and adds that it *refers to all of the actions required to maintain access to digital materials beyond the limits of media failure or technological change.*[JB02] It is confined from digitisation, which is a challenging field in itself.

2.1.1 The Current State of Research

At the moment libraries, archives and scientific institutions as well as government agencies, large industries, SMEs and also private persons, who have steadily growing amounts of legally or personally important data, are increasingly facing the problems of long-term archival and access. This resulted in the creation of a number of large-scale initiatives integrating digital preservation capabilities in digital repository systems [Smi05, TBS$^+$03].

A good overview of preservation of strategies for the preservation of digital heritage is provided by the companion document to the UNESCO charter for the preservation of the digital heritage [Web05]. Rosenthal [RRL$^+$05] describes requirements for ideal digital preservation systems.

Many worldwide projects and initiatives deal with the challenge of digital preservation and providing services and information. Because of the enormous increase in interest, the number of projects is growing rapidly. The following section can thus only present an incomplete overview, which is not meant to be an assessment.

The National Digital Information Infrastructure and Preservation Program (NDIIPP) collects, archives and preserves the burgeoning amounts of digital content for current and future generations.[1]

The Arts and Humanities Data Service (AHDS) and the University of London Computer Centre started the DAAT Project (Digital Asset Assessment Tool) [Art06] to develop a tool to identify the preservation needs of various digital holdings.

PREMIS (Preservation Metadata Implementation Strategies) is a joint effort of the Research Libraries Group (RLG) and the library consortium OCLC (Online Computer Library Center). The aim of the project is the development and recommendation of best practices for implementing preservation metadata.[2]

PANIC[3] (Preservation webservices Architecture for Newmedia and Interactive Collections) addresses the challenges of integrating and leveraging services and tools into a Preservation Services Architecture. Also the comparison of different preservation strategies for multimedia data and the devel-

[1] http://www.digitalpreservation.gov

[2] http://www.loc.gov/standards/premis

[3] http://www.metadata.net/panic

opment of recommendations and guidelines for multimedia content are part of PANIC [HC06, HC05, HC04].

CAMiLEON is developing and evaluating a range of technical strategies for the long term preservation of digital materials. The project is a joint undertaking between the Universities of Michigan (USA) and Leeds (UK) and is funded by JISC and NSF.[4]

PADI (Preserving Access to Digital Information) is an initiative started by The National Library of Australia that aims to provide mechanisms to ensure that information in digital form is managed with appropriate consideration for preservation and future access.[5]

The LOCKSS[6] program (Lots of Copies Keep Stuff Safe) has developed a peer-to-peer system using the world wide web for preserving archival information. The first version was launched in 1999. LOCKSS is an open source software that deploys a large number of independent, low-cost, persistent and accessible Web caches. It allows the peers to cooperatively detect and repair damaged content. For obsolete file formats LOCKSS supports a format migration[RLRS05].

The Digital Curation Centre (DCC) supports UK institutions in storing, managing and preserving data. The aim of the project is to establish community relationships and develop services for digital preservation.[7]

The Digital Preservation Coalition (DPC) aims to secure the preservation of digital resources in the UK. DPC hosts the Digital Preservation Award to bestow leading and advanced digital preservation projects and, in cooperation with PADI, publishes a quarterly current awareness digest.[8]

The Electronic Resource Preservation and Access Network (ERPANET) established a European Consortium bringing together memory organisations (museums, libraries and archives), ICT and software industry, research institutions, and government organisations to provide a knowledge-base of developments in the area of preservation of cultural heritage and scientific digital objects.[9]

Several projects in this domain have been initiated under the 6th framework program of the European Union. Two of them are of particular interest for this thesis: DELOS and PLANETS.

One of the main objectives of the DELOS Digital Preservation Cluster,which is part of the EU-funded DELOS Network of Excellence on Digital

[4] http://www.si.umich.edu/CAMILEON

[5] http://www.nla.gov.au/padi

[6] http://www.lockss.org

[7] http://www.dcc.ac.uk

[8] http://www.dpconline.org

[9] http://www.erpanet.org

Libraries[10], is the establishment of testbeds and validation metrics. The DELOS Digital Preservation Testbed [SRR$^+$06] allows the selection of the most suitable preservation strategy for individual requirements by combining a structured workflow for requirements specification and evaluation by means of a standardised testbed laboratory infrastructure.[11]

PLANETS (Permanent Long-term Access through Networked Services) develops systems and tools which support the accessibility and use of digital cultural and scientific resources. More specifically, the project is developing methods and tools based on a distributed service infrastructure on which services for preservation action, preservation characterisation and preservation planning can be coordinated and combined with each other. The PLANETS Testbed will use this framework to provide a stable foundation for evaluating different preservation actions in a well-defined setting.[12]

Digital Preservation Europe (DPE) is a coordinated Action of the EU 6th framework program bringing together leading institutions and researchers to provide a coherent platform for research and collaboration. Aspects covered by DPE include audit and certification, and the establishment of a research roadmap. DPE will conduct a yearly Digital Preservation Challenge with the aim to raise the profile of digital preservation among researchers and foster awareness of the issues the community is facing. DPE initiated the Digital Preservation Europe Exchange Program to support the exchange of researchers in the field of digital preservation.[13]

Another project funded by the EC, CASPAR (Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval), is targeted at implementing, extending, and validating the OAIS reference model, but also addresses issues such as digital rights management and intends to design virtualisation services supporting long-term preservation.[14]

A number of tools and services have been developed that help in identifying and describing file formats.

The National Library of New Zealand Metadata Extraction Tool[15] extracts preservation metadata of a range of file formats. JHove [Har05], developed by JSTOR and the Harvard University Library, supports the identification and characterisation of digital objects. These tools can be used to analyse files of migration experiments. File format repositories, such as PRONOM [Pet03] may be used to identify specific characteristics of the digi-

[10] http://www.delos.info

[11] http://www.dpc.delos.info

[12] http://www.planets-project.eu

[13] http://www.digitalpreservationeurope.org

[14] http://www.casparpreserves.eu

[15] http://www.natlib.govt.nz/en/whatsnew/4initiatives.html\#extraction

tal objects at hand, helping in the elicitation of the preservation requirements. Currently the fourth version of PRONOM is available. This initiative also develops an automatic file format identification tool named DROID (Digital Record Object Identification).[16]

2.1.2 Approaches to Digital Preservation

Research on technical preservation issues and actual solutions is focused on two dominant strategies – migration and emulation.

Migration

Migration refers to the conversion of a digital object to another representation. This might be a newer version of the same file format or a different format. Migration thus operates on the objects themselves in order to transform them to a more stable form that is better suited for long-term access. For example, text documents written in MS Word are often migrated to Adobe PDF because it is considered to be more stable.

The Council of Library and Information Resources (CLIR) presented different kinds of risks for a migration project [LKR$^+$00]. The critical problem generally is how to ensure consistency and authenticity and preserve all the essential features and the conceptual characteristics of the original object whilst transforming its logical representation. Each migration step incurs certain risks and preserves only a certain fraction of the characteristics of any digital document. The conversion of a text document to PDF as mentioned above, for instance, changes its behaviour and the look-and-feel the user experiences. Moreover, metadata such as edit history are lost. Converting a document with complex formatting often poses challenges even to the best tools. On the other hand, migrating these documents to images preserves the layout, but loses the machine-readable textual content. Even migrations within the same format family may incur unwanted and unspecified changes. Still, the number of available tools as well as the ease of applying this strategy and the possibilities for large-scale usage make it a promising candidate.

Emulation

In contrast to migration, emulation as the second important preservation strategy operates on environments for objects rather than the objects themselves. Emulation aims at mimicking a certain environment that a digital object needs, e.g. a certain processor or a certain operating system.

[16] `http://www.nationalarchives.gov.uk/pronom`

Jeff Rothenberg [Rot99] envisions a framework of an ideal preservation surrounding for emulation. Several projects have been working on the development of emulation approaches. The BCC Domesday project as proof-of-concept case study for the CAMiLEON project [Mel03] implemented an emulation strategy to preserve access to the digitised version of a 900-year-old Britain book called the Domesday book.

Reichherzer [RB06] reports on large scale emulation to provide access to archived office documents. The Digital Asset Preservation Tool as an implementation of the Universal Virtual Computer (UVC) is described in [HVDDVEM05], [Lor02] and [WO04]. So far the tool supports the file formats JPEG and GIF87a. Van der Hoeven presented an emerging approach to emulation called *Modular emulation* in [vdHvW05].

A few other preservation strategies have been proposed. The Computer Museum is based on preserving the technical environment that runs the system, such as the hardware, the operating system, original application software, and media drives. Normalisation is a technique collecting files in different formats and converting them to a single file format [CLI02].

2.1.3 Preservation Planning

A range of tools exists today to support the variety of preservation strategies such as migration or emulation. Yet, different preservation requirements across institutions and settings make the decision on which solution to implement very difficult.

Preservation Planning, i.e. evaluating preservation strategies and choosing the most appropriate strategy, has turned into a crucial decision process, depending on both object characteristics as well as institutional requirements. The selection of the preservation strategy and tools is often the most difficult part in digital preservation endeavours. Technical as well as process and financial aspects of a preservation strategy form the basis for the decision on which preservation strategy to adopt.

Figure 2.1 provides an overview of the workflow of the preservation planning procedure adopted in PLANETS, which is based on the DELOS Digital Preservation Testbed. The DELOS workflow was described in [RR04] and recently revised and described in detail in [SRR$^+$06]. During the PLANETS project, it has been refined to reflect feedback gathered during a series of case studies. The process consists of three phases, which are described in the following.

1. **Define requirements** describes the scenario, the collection that is being considered as well as institutional policies and obligations. Then

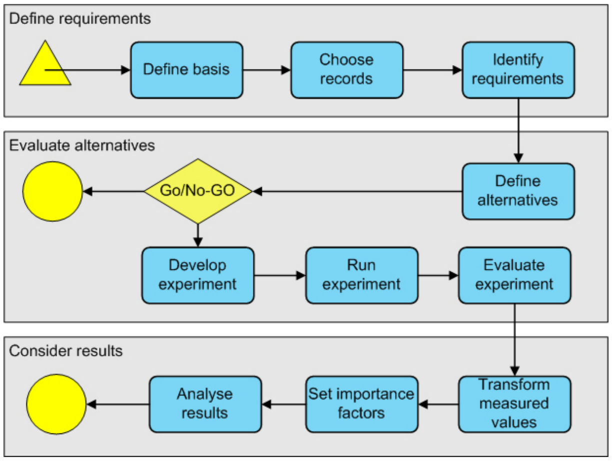


Figure 2.1: Overview of PLANETS preservation planning workflow

the requirements and goals for a preservation solution in a given application domain are defined. In the so-called objective tree, high-level goals and detailed requirements are collected and organised in a tree structure.

While the resulting trees usually differ through changing preservation settings, some general principles can be observed. At the top level, the objectives can usually be organised into four main categories:

- *File characteristics* describe the visual and contextual experience a user has by dealing with a digital record. Subdivisions may be "Content", "Context", "Structure", "Appearance", and "Behaviour" [RB99], with lowest level objectives being e.g. color depth, image resolution, forms of interactivity, macro support, or embedded metadata.
- *Record characteristics* describe the technical foundations of a digital record, the context, the storage medium, interrelationships and metadata.
- *Process characteristics* describe the preservation process. These include usability, complexity or scalability.
- *Costs* have a significant influence on the choice of a preservation solution. Usually, they may be divided in technical and personnel costs.

The objective tree is usually created in a workshop setting with experts from different domains contributing to the requirements gathering process. The tree documents the individual preservation requirements of an institution for a given partially homogeneous collection of objects. Examples include scientific papers and dissertations in PDF format, historic audio recordings, or video holdings from ethnographic studies. Typical trees may contain between 50 to several hundred objectives, usually organised in 4-6 hierarchy levels.

Measurable effects are assigned to the objectives that have been defined in the previous step. Wherever possible, these effects should be objectively measurable (e.g. € per year, frames per second). In some cases, (semi-) subjective scales will need to be employed (e.g. degrees of openness and stability, support of a standard, degree of file format adoption, etc.).

2. **Evaluate alternatives** identifies and evaluates potential alternatives. The alternatives' characteristics and technical details are specified; then the resources for the experiments are selected, the required tools set up, and a set of experiments is performed. Based on the requirements defined in the beginning, the results of the experiments are evaluated to determine the degree to which the requirements defined in the objective tree were met.

3. **Consider results** aggregates the results of the experiments to make them comparable. The measurements taken in the experiments might all have different scales. In order to make these comparable, they are transformed to a uniform scale using transformation tables. The resulting scale might e. g. range from 0 to 5. A value of 0 would in this case denote an unacceptable result and thus serve as a drop-out criterion for the whole preservation alternative.

   Then the importance factors are set, as not all of the objectives of the tree are equally important, and the alternatives are ranked. The stability of the final ranking is analysed with respect to minor changes in the weighting and performance of the individual objectives using Sensitivity Analysis. The results are finally evaluated by taking non-measurable influences on the decision into account. After this analysis a clear and well argumented accountable, recommendation for one of the alternatives can be made.

Ongoing work in PLANETS aims at implementing this workflow and integrating it within the distributed service network of PLANETS which we will outline in Section 3.2.

2.2 Testing and Evaluating Digital Preservation Strategies and Tools

A wide range of tools exist nowadays to support digital preservation activities. For converting office formats to PDF alone, one could use a dozen of different conversion tools ranging from freely available single-file tools such as MyMorph[17] or BullZip Printer[18] to expensive server solutions capable of mass-migrating thousands of files such as, e.g., the Adlib ExpressServer[19].

These tools have very specific characteristics and capabilities. As every institution, be it a national archive, a research library, a large company or a government agency, has specific technical requirements as well as such related to the technical environment and the organization, a thorough investigation of the characteristics and thus the applicability and fitness for purpose of each tool is inevitable before any commitment in an organization is made.

The preservation planning approach described in Section 2.1.3 with its detailed requirements specification is of great value in this regard. However, for investigating in detail technical characteristics like performance values for mass-migrating large numbers of digital objects, a controlled and standardized environment is needed, a dedicated hardware and software platform with a set of benchmark corpora on which controlled and well-defined experiments can be run to compare properties of different tools, configuration settings and input data across institutional boundaries.

This section will lay out the foundations of such a controlled environment for testing systems, which in the digital preservation domain is called a *testbed*.

System Testing as described in [Ger91, Bla02] is a form of *black-box testing* which evaluates a system's compliance with its specified requirements without inspecting the interior parts of this system. Thus it needs no knowledge about the inner design of the system under consideration, as opposed to white-box testing, which relies on knowledge about the inner structure of the system that is being tested. Instead, black-box testing views the entire system as one unit. Examples for characteristics that may be tested include

- Usability testing,
- Security testing,
- Capacity testing,

[17] http://docmorph.nlm.nih.gov/doc

[18] http://www.bullzip.com

[19] http://www.adlibsoftware.com/ExpressServer.aspx

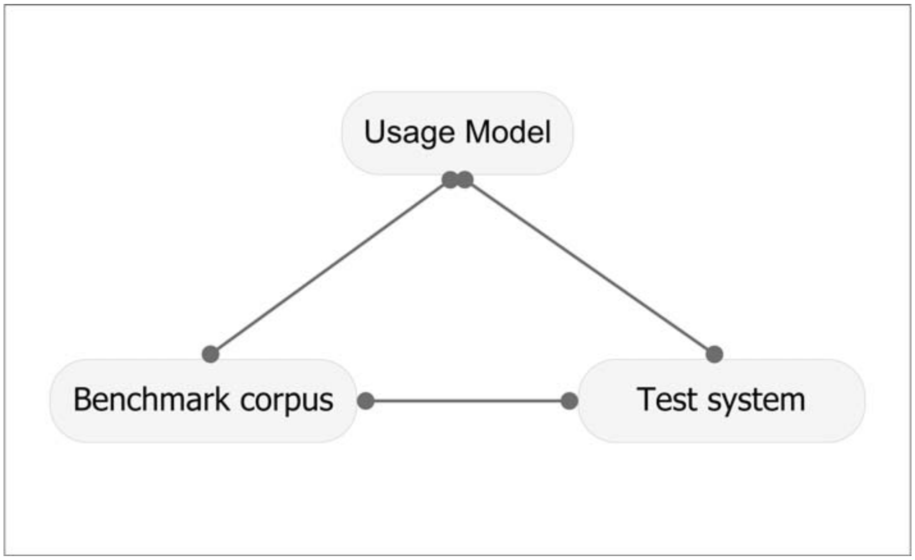


Figure 2.2: The three core building blocks for testing digital preservation solutions

- Performance testing, and
- Reliability testing.

2.3 Core Building Blocks of a Digital Preservation Testbed

Three aspects form the basis for a controlled environment for testing digital preservation solutions, as depicted in Figure 2.2.

1. Well-defined, possibly domain-specific **benchmark corpora**,
2. A **test system** and platform, and
3. A **well-defined usage model**.

This work will focus on the third core building block, the usage model of a testbed platform. The following sections give an overview on benchmark data sets and requirements on the test platform. After an introduction to the use case methodology, which is used for requirements definition, the following chapter will present the Dutch Testbed and lay out the motivation for extending it within the PLANETS project. Chapter 4 presents the functional model for the PLANETS Testbed.

2.3.1 Benchmark Data Sets for Digital Preservation

Task 6.9 of the DELOS Digital Preservation Cluster is defining a *substantial, consistently designed and documented* benchmark corpus for digital preservation [NBL+07]. The report that is being written as part of this effort forms the basis for this section.

Introduction

Taking into account the variety of institutions involved, the range of application domains, the diversity of object and file formats involved and the legal and organizational issues further adding to the complexity, we have to acknowledge that building a corpus for digital preservation is a challenging task.

The following two aspects of system evaluation are most relevant in our context:

- **System quality evaluation** measures the correctness and appropriateness of applied solutions with respect to the quality of the outcome by comparing the results of a given solution to an accepted standard method, which has to be identified first.

- **System performance evaluation** should cover both performance per object and scalability issues.

The collections that form part of the benchmark have to satisfy specific criteria. For benchmarking machine-learning algorithms, digital objects have to be tagged; for scalability experiments, the collection has to be sufficiently large. In the digital preservation domain, this is of major importance as most of the applications will likely have to be applied to huge numbers of digital objects and a broad range of different institutional backgrounds.

Corpora can in general be categorized according to their purpose:

A **content-complete** corpus covers the widest variety of possible types of content available in a given scenario. The purpose of such a corpus is the testing of organizational procedures: For instance, all the digital object types used in a given organization or setting are covered, in order to describe their specific problems.

In contrast to this, a **feature-complete** corpus is defined by the coverage of the widest variety of possible features of a given object type. Such a corpus therefore is, by definition, object-type specific, and can be used to test the completeness of implementations for a given object type.

The DELOS report further defines a **performance-defining** corpus as containing *set of objects which is sufficiently large so that two or more programs processing it can be compared in a meaningful way with respect to their performance.*([NBL$^+$07], p. 6).

Usage Scenarios in Digital Preservation

A wide range of different application scenarios can be identified in digital preservation, since every content holding institution has different collections and user communities. Application scenarios and requirements include

- State and National Archives, having a strong focus on authenticity;
- Publications of the academic community, available mainly as object types like `.pdf`, `.ps`, or `.doc`;
- Scientific data, particularly from the natural sciences, with the principal characteristic of being very large, highly domain-specific and in proprietary formats; and
- Collections of interactive and multimedia art pose highly challenging requirements both because of the complex characteristics of these objects and because of the many proprietary data formats that are used.

In the following we will outline two potential scenarios for DP benchmarking which are of particular interest because of their specific peculiarities.

Archival Records in Governmental Archives Government agencies produce during the course of their activities a huge amount of information in a vast range of object types and formats ranging from simple text documents to complex databases and websites or even highly complex GIS systems. Especially for accountability reasons this information needs to comply with requirements such as authenticity, integrity and reliability. The abovementioned report defines the authenticity of records as determined by

> 1. the (intellectual and technical) characteristics of records existing at the moment they were created and used to perform their function in the business context, and
> 2. the ability to identify how they were and are managed after they have been captured under a recordkeeping regime.[NBL$^+$07]

Government records are almost always aggregated into case files, series of records or other aggregations. The relationships between these records need to be preserved as they are essential for understanding their content, and they are also part of the authenticity of records. Most of these requirements result in a variety of metadata that need to be preserved. The main challenge is to maintain all these metadata associated with the records, including their interrelationships, through time.

Collections of Interactive and Multimedia Art The challenge of preserving born-digital multimedia art, which is inherently interactive, virtual, and temporary, has been an actively discussed topic in the last years. In 2004, the ERPANET project organised a workshop [ERP04] on archiving and preservation of born-digital art. Preserving the inherent complexities of interactive multimedia is a very difficult task, particularly because formats used in multimedia art are ephemeral and unstable. It also poses a paradox between the transformation necessary to keep the work accessible, and desired authenticity of each piece of art.

Some modern museums hold large numbers of born-digital art pieces. The Ars Electronica Center (AEC) in Linz, Austria[20] has been collecting electronic art in digital form since the early nineties. The AEC holds more than 25.000 CDs containing multimedia and interactive art in different formats like long-obsolete presentation file formats with interactive visuals, audio and video content. The Ludwig Boltzmann Institute[21] is currently evaluating alternative strategies to not only preserve these pieces of art over the long term, but also make them accessible in a satisfying form on the web.

These collections pose extreme problems to digital preservation due to their specific and complex characteristics.

1. The collections are highly heterogenous, there is no common file format. Instead, digital art ranges from rather normal image and video files to specifically designed, proprietary software pieces which are sometimes highly dependent on a specific environment. Some pieces of art even deal with the issues of digital deterioration and damaged content by presenting image files with destroyed regions.

2. Often artists object to the idea of preserving their artwork, because they feel its value lies in the instantaneous situation, it should be volatile or they want to retain control about the original object.

All of the above considerably adds to the complexity of constructing a corpus of digital multimedia art and makes it all the more important to do it thoroughly and precisely.

[20] `http://www.aec.at/en/center`

[21] `http://media.lbg.ac.at/en/institution.php?iMenuID=1`

Current case studies conducted in a joint effort of the Institute for Software Technology and Interactive Systems at the Vienna University of Technology[22] and the Boltzmann Institute as part of the PLANETS project should bring first perspectives on the requirements for such a corpus.

Further input might come from the Variable Media Network and the collaborative project 'Archiving the Avantgarde' [bam07] undertaken by the Berkeley Art Museum and Pacific Film Archive (BAM/PFA), the Solomon R. Guggenheim Museum, the Walker Art Center, Rhizome.org, the Franklin Furnace Archive, and the Cleveland Performance Art Festival and Archive. The Guggenheim Museum has also established the Variable Media Initiative [var07], now known as the Variable Media Network, which has invited media artists, curators, and museum specialists to a series of meetings to brainstorm strategies for preserving specific case study works in the Guggenheim collection.

Existing Corpora in Information Retrieval

In several disciplines of computer science, benchmarking corpora have been created. The most relevant domain in this context is information retrieval [SM86].

Text Retrieval – Two well-known corpora stand out in the text retrieval domain:

- The three **Reuters news corpora** reuters21578 [Lew], Reuters Corpus Volume 1 (RCV1, [RSW02]) and RCV2 [RSW05] contain new stories distributed in plain text and XML. The reuters21578 contains 21578 news stories coming from a wide range of different domains. RCV1 contains 810.000 messages in English, while RCV2 consists of 487.000 stories in thirteen different languages.

- The benchmark corpora of the **TREC Text Retrieval Conference** is another prominent example [Nat]. The conference organises yearly tracks for specific tasks. These tasks include, e.g.,

  - question answering[23], where users may formulate queries in natural language and the answer is not a list of documents matching the query, but rather a sentence that should be formulated as naturally as possible; and

[22] http://www.ifs.tuwien.ac.at/dp

[23] http://trec.nist.gov/data/qa.html

  - email spam filtering[24], where participants submit a spam filter for email in different languages.

  In 2001 and 2002, the conference also contained a video track, which became in independent evaluation (TRECVID)[25] in 2003.

  Parts of the data used in these tasks are in the public domain. The TREC 2005 Terabyte track used a web archive harvested from sites in the `.gov` domain which consisted of 25 million documents.

Music Information Retrieval – Publicly available corpora pose a big challenge in the domain of music information retrieval due to copyright restrictions that apply for most interesting collections.

The most prominent example for corpora in this area are the ones used by the International Conference of Music Information Retrieval (ISMIR). This conference hosts a yearly contest called MIREX which both symbolic and audio-based music information retrieval and includes tasks such as

1. Genre classification,
2. Melody extraction,
3. Key finding, or
4. Symbolic melody similarity.

The benchmark data used in the MIREX contest are not publicly available due to the copyright restrictions mentioned above. Instead, participants have to submit their code, which runs on the site hosting the corpus. Participants have no knowledge about the data used for evaluation.

Challenges on Digital Preservation Corpora

Ongoing work in the DELOS project has identified five main challenges for corpus generation in the digital preservation context.

1. **Size.** – The size of a corpus refers to both the number of objects contained and the actual file or corpus size in terms of storage space needed.

[24] `http://plg.uwaterloo.ca/~gvcormac/spam`

[25] `http://www.itl.nist.gov/iaui/894.02/projects/trecvid`

2. **Stratification** means the coverage of types of digital objects and domains needed by a specific user community, i.e. the distribution of elements according to specified criteria. Potential categories for stratification include file type, time, or real-world scenarios.

3. **Data Representation** is a big issue across domains, as complex digital objects need to be transformed to a common form to be able to compare results.

4. **Precise Task Definitions** are of vital importance to ensure the usefulness of a corpus for digital preservation in practice.

5. **Ground Truth** denotes the criteria against which evaluation takes place, which are often determined by human evaluators. These annotations can then be used to judge the correctness of systems. Defining a ground truth for digital preservation corpora is inherently difficult and often even impossible because the question which preservation solution is the ideal one always depends on the specific context and the requirements of each institution. Instead of defining one ground truth, it would be possible to define a framework allowing every institution that uses the corpus to find its optimal solution. A starting point for this might be the case studies on preservation planning and the essential characteristics of specific collections that have been undertaken as part of the DELOS Testbed and the PLANETS project. Results of these studies can be found in [SBN+07, SRR+06, SBNR07]. The PLANETS project is conducting further studies, among others dealing with collections of multimedia art, image collections, and web archives. These case studies will in the future lead to generalised requirements definitions for digital preservation strategies which might be suited for this task.

Object Types and File Formats

As mentioned before, stratification is a necessity for the creation of benchmarking corpora. Particularly for a 'content complete' corpus which is supposed to the widest variety of content, stratification along file and object types, respectively a limitation thereof, is essential since no corpus will be able to hold all existing object types. The number of known object types or file formats is tremendously high, FILExt lists over 20.800 object type entries [Com02]. Complete coverage therefore seems to be unrealistic. The Pronom registry [Arc] also lists more than 500 categories of object types. Studies on the diversity of types which are actually used show that most content is covered by a small fraction of existing object types. For the Danish Internet domains, 95 per cent of the collected data is covered by 15 file types [And05]. This study also found out that most of the harvested files had no MIME type, the official categorisation of object types, associated, making that categorisation a not perfectly feasible choice. Very similar results are also shown in [RAW02] as well as [Koe05]. The Library of Congress released a proposal for the most relevant object types for digital preservation [Lib]. Further, the Florida Centre for Library Automation suggests relevant object types in [Flo05]. Another categorisation, the 'file format encyclopaedia', is presented in [Mil], albeit slightly outdated. A comprehensive overview is given in the 'File formats typology and registries for digital preservation' deliverable of the DELOS project [GC04].

Summary and Outlook

While building a corpus for digital preservation is a highly challenging task, the significant progress that is being made in the DELOS project shows that the digital preservation community might be able to build upon open digital object corpora within a few years. These corpora that are still to be built will form a highly valuable building block for testing digital preservation strategies.

2.3.2 Test Platform

The second important part of a digital preservation testbed is a dedicated test system and platform to run the experiments on. This system has to be equipped with

1. sufficiently large **storage facilities** to enable participatory corpus building and the permanent storage of large amounts of test data, and
2. powerful **processing capabilities** to handle large-scale experiments such as the migration of tens of thousands of files within reasonable time.

In the PLANETS project, the main instance of the testbed will be deployed within the hardware infrastructure of the University of Glasgow and will be managed by the Humanities Advanced Technology And Information Institute (HATII). It will run on a dedicated SUN V240 server running Solaris 10. The server will contain two 1.5GHz Sparcv9 processors and 8GB of memory and server will be connected to a Storage Area Network that will provide at least nine terabytes of dedicated storage space.

Further instances of the software platform may be installed by the partner institutions as needed.

2.3.3 Usage Models: The Use Case Methodology

We employ the use case methodology to describe the functional requirements for the testbed. The concept of use cases emerged in the early nineties and is discussed extensively in several textbooks [JCJv93, Coc00, OP04]. Cockburn describes this concept as follows:

> A use case captures a contract between the stakeholders of a system about its behaviour. The use case describes the system's behaviour under various conditions as the system responds to a request from one of the stakeholders, called the *primary actor*.[26]

Similarly, Övergaard et. al. define use cases as

> ...defining how the modeled system is to be used by its surroundings. A *use case* models one usage of the system; that is, it describes what sequences of actions will be performed by the system as a response to events occuring outside the system caused by the users.[27]

Thus, a use case is a textual description of the behaviour of a system in reponse to outside events caused by the system users. Use cases have been applied successfully in a vast variety of fields, ranging from embedded systems [NMB02] and client-server supply-chain simulation [CBC$^+$99] to B2B e-commerce applications [SJP02]. They are not only successfully being applied for systems analysis, but also in areas like business process reengineering [JEJ94, Coc00]. The textual character is one of the main advantages of use cases over more formal methods. They foster communication between stakeholders with varying backgrounds and different levels of technical expertise.

A use case may be written in different forms and may contain several sections of text; which ones are actually contained varies greatly between projects and depends on a number of parameters like

- the style of writing,
- the intended audience,

[26] [Coc00],p. 1

[27] [OP04], p. 35

- organization-wide policies, and
- the requirements of the project.

Some of the most common sections that may be present in a use case description are

- The **name** of the use case.
- The **scope** defines what is considered black box, i. e. the system under design.
- **Level** should be one of summary, user-goal, or subfunction.
    1. A **summary level** use case describes a (probably longer-term) overall goal of a user, like *Sell an item* in an online auctioning system.
    2. A use case at the **user-goal** level describes a necessary step for achieving a summary level goal, e. g. *Register user.*
    3. A **subfunction** use case in turn describes a lower-level goal that does not yield any direct benefit to the primary actor; it is necessary to complete a higher-level step. A common example is *Logon user.*
- The **Primary Actor** is the stakeholder requesting a service from the system in order to achieve a goal. Usually, the primary actor is the one triggering the use case.
- **Intent** captures the goal of the actor in a short statement.
- **Context of Use** may consist of a statement of the goal and possibly the circumstances under which this use case occurs.
- **Secondary or Supporting Actors** are other systems that are supposed to deliver subgoals for the system under design.
- **Preconditions** state the conditions that have to be met before the use case can execute. The most common example is *User is logged on.*
- **Minimal Guarantees** state what the system guarantees to be achieved even in case of a failure during the flow of events. Some common minimal guarantee would be *The system logs the progress of the use case execution.*

- **Success Guarantees** state what is achieved if the use case executed successfully. For the *Logon* use case, this might be *User is successfully authenticated and session initialized.*

- **Trigger** describes the event causing the use case to start. For a use case *Handle emergency call* in an emergency operation call center, this might be *Someone calls 112.*

- **Includes** states which use cases are included, or used, within the flow of this use case. E. g., a use case *Edit Something* may include a use case *Search for Something.*

- The section **Flow of events**, also named **Main Success Scenario** or simply **Description**, contains the main eventflow, the description of a series of steps that comprise a successful execution of the use case. Errors and alternative flows are usually handled separately, in the next section.

- **Extensions** or **Exceptions** define alternative flows of events, e. g. when conditions cannot be validated successfully. If, for instance, in the above mentioned use case *Logon user* some steps of the main success scenario read *4. The system verifies the provided credentials... 5. The sytem grants access*, a possible exception would be *4a. The supplied credentials are not correct: The user is notified and may enter the credentials again.*

- **Management information** like priority, due date, history of changes, responsible author, frequency of occurence, etc., is sometimes attached to the use case description.

Cockburn suggests (among others) two different styles of writing a use case:

1. The **casual use case** form is often used in agile environments or during early stages of a software project. It contains only a few sections, e. g.

   (a) Primary Actor,

   (b) Scope,

   (c) Level, and

   (d) the description of the eventflow.

2. A **fully dressed use case**, on the contrary, features a much more thoroughly structured way of description; it may consist of a lot more sections like those mentioned in the listing above, and will usually use numbered steps to describe the flow of events and the extension list.

Furthermore, he suggests a simple way of employing some of the benefits of use case writing while at the same time investing minimum time, effort and standardization: Writing *use case briefs* that contain as little as the sections described in the casual case above or even less, with the description being just a few lines. These use case briefs can be of benefit especially in agile environments where communication between team members is very tight.

Of course, in reality one finds use case styles everywhere in between the range of variations described above. For specifying the functional model in Chapter 4, we will use the following sections:

1. Name,
2. Scope,
3. Level,
4. Actors,
5. Preconditions,
6. Trigger,
7. Success Guarantees,
8. Main Success Scenario, and
9. Exceptions.

2.4 Summary

In this chapter, we introduced the main concepts and foundations that underly this work. We gave an overview of digital preservation, its challenges, and the work that has been undertaken so far to tackle the main problems.

We then introduced the issue of testing and evaluating digital preservation strategies. An indispensable requirement for the consistent evaluation and comparison of digital preservation strategies in an objective and repeatable manner is the existence of a benchmark corpus. We introduced efforts for defining data corpora for benchmarking purposes and describe the specific challenges for creating corpora for digital preservation.

The chapter concluded with an introduction on the use case methodology that we are employing for the specification of the functional model of the testbed.

In the next chapter, we will report on the Dutch Testbed, which can be seen as the predecessor of the PLANETS Testbed. We will discuss its main characteristics and its shortcomings and explain the goals and characteristics of the PLANETS project in general and the PLANETS Testbed in particular.

Chapter 3

A Digital Preservation Testbed

This chapter introduces the testbed devised in the Dutch National Archives, which forms the basis for the PLANETS Testbed. We discuss the workflow and the testbed laboratory and outline the basic architecture of the testbed system.

We then introduce the PLANETS project and its main architecture, and lay out the main goals of the PLANETS Testbed and the core tasks that the testbed software needs to support.

3.1 The Dutch Testbed

3.1.1 Research Objectives

The Testbed project was a three-year research project that started in September 2000. Its main objectives were to provide insight into the following aspects of digital preservation:

- **Authenticity features of digital objects.** – An authentic object as defined in [HVD$^+$04] is *an object than can be proven to be what it purports to be and to have been created (or sent) by the person purported to have created (or sent) it.*

- **Management processes and activities concerned with metadata** that support all phases in a digital archive, ranging from ingest and preservation to access of digital records.

- **Cost factors** of the storage and management of digital records and their metadata

- **Technical solutions** for digital preservation and their effectiveness.

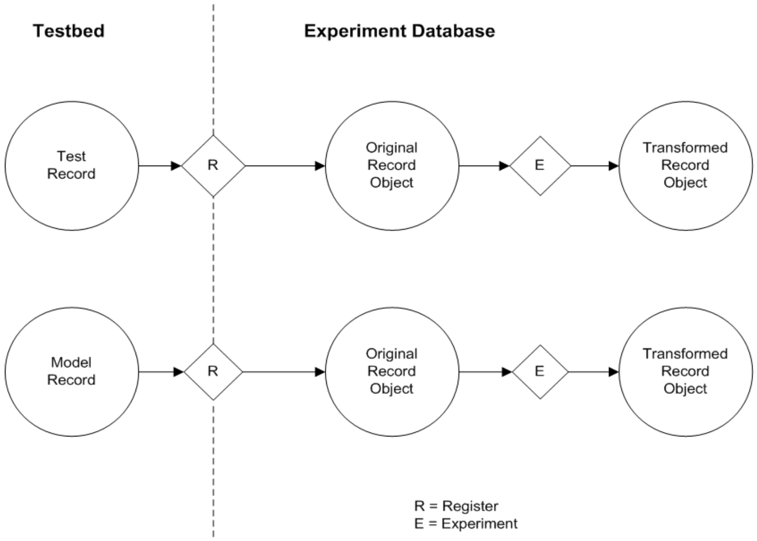


Figure 3.1: Test records [otN01]

3.1.2 Conceptual Framework

To achieve these objectives, the Testbed project constructed a digital laboratory environment for conducting digital preservation experiments that evaluate alternatives for the preservation of digital records in a way that ensures authenticity. The approach is described in detail in [otN01] and [HVD$^+$04]. Slats and Verdegem report on practical findings of the project in [SV04]. We will continue by outlining the main concepts, the organisational workflow and the laboratory environment of the Dutch Testbed.

The Testbed Laboratory is a hardware and software infrastructure, designed to support the research of the Digital Preservation Testbed project. It is clearly not a repository system like the Digital Archive System of the National Archives of the UK [Art06] or the e-Depot system of the National Library of the Netherlands [OvDvW04]. Instead, it is complementary to these systems in that it addresses what is needed to do to maintain sustainable long-term access to digital objects. The whole system is based around the concepts of experiments and records. Test records and model records are temporarily stored in the Testbed Laboratory until the records are registered. Model Records are created by the Testbed project and are intended to provide a stable basis of evaluation, they can be seen as a benchmark data set. Test Records are provided by participating partners as real examples. Registration will associate a basic set of metadata with the test and model records for the purpose of identification and control and thus produce Original Record Objects as depicted in Figure 3.1. The experiments consist of transforming Original Record Objects (ORO) to Transferred Record Objects (TRO).

The process of defining, executing and evaluating experiments follows a defined workflow with clearly specified roles and responsibilities as shown in Figure 3.2 and outlined in the following.

1. **Define exploration area** is a rather informal step of deciding which aspects of digital preservation to explore in the experiment.

2. **Prepare Experiment** includes background studies concerning e.g. technical properties of the records to be considered. Any information that is gathered in this stage is captured and accessible through the research database.

3. **Define Requirements** determines the requirements, i.e. success criteria upon which evaluation will be based. In the Dutch Testbed these are primarily focused on *authenticity*.

4. **Develop Experiment Design** defines

    - the experiment purpose and the research questions that it addresses,
    - the types of records that will be used,
    - the complete procedures and metadata needed for setting up, running, and evaluating the experiment,
    - the requirements of the software to be developed, and
    - the type of results expected and the validation to be performed on the results.

5. **Specify Resources** provides an estimate of the time and resources needed to prepare, test, and perform the experiment, including possible software development effort.

6. The **Go/No Go Decision** investigates the estimate produced in the previous stage and considers the potential benefits of the experiment. Then a documented decision is taken whether to continue, change the course of the experiment or abandon it.

7. **Develop Experiment** produces a specific plan for testing and running the experiment, and also develops any specialised software needed for the experiment.

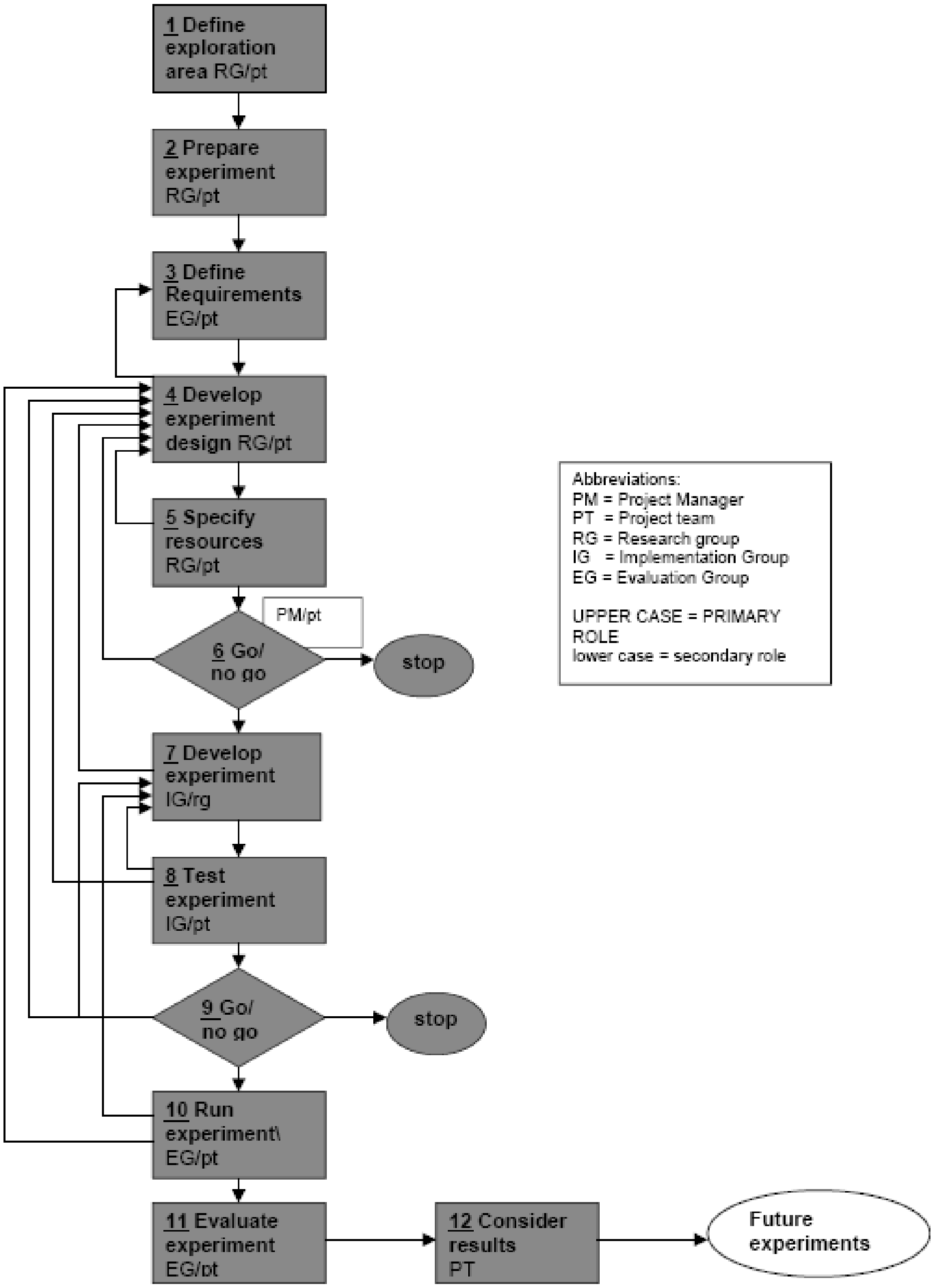

1 Define exploration area RG/pt
2 Prepare experiment RG/pt
3 Define Requirements EG/pt
4 Develop experiment design RG/pt
5 Specify resources RG/pt
PM/pt
6 Go/ no go
stop
7 Develop experiment IG/rg
8 Test experiment IG/pt
9 Go/ no go
stop
10 Run experiment\ EG/pt
11 Evaluate experiment EG/pt
12 Consider results PT
Future experiments
Abbreviations:
PM = Project Manager
PT = Project team
RG = Research group
IG = Implementation Group
EG = Evaluation Group
UPPER CASE = PRIMARY ROLE
lower case = secondary role

Figure 3.2: Dutch Testbed workflow [HVD+04]

It is particularly noteworthy that this step specifically includes the notion of developing software specifically for an experiment, because very often the software needed is not yet available. This is one of the aspects that the PLANETS Testbed tries to change by delivering and connecting a range of services in a distributed environment.

8. **Test Experiment** systematically tests running experiments and reviews the results in order to identify aspects that need to be revised or further developed.

9. A second **Go/No Go Decision** consists of an analysis of the results of the previous stages to determine if the experiment should be delayed or cancelled or if it should proceed as planned.

10. **Run Experiment** finally produces results, which could be converted files, revised metadata, etc., to be evaluated in the next step.

11. **Evaluate Experiment** determines how successfully the requirements were met. This step also considers further experiments that may be needed, possibly on other records, and secondary results that can be developed, such as policies or guidelines. An evaluation can also compare the results of different experiments.

    In order to provide stable circumstances for identifying differences between the original record and the transformed record, the experimental evaluation is to be run in a standardised laboratory setting.

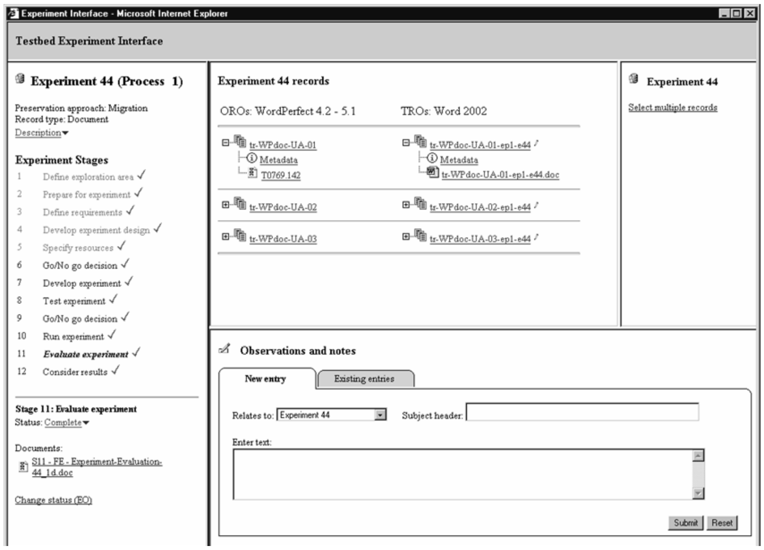
12. **Consider Results** analyses all stages to make recommendations for future experiments and provide evaluation of the testbed itself.

The workflow is supported by a software guiding the process, the interface of which is shown in Figure 3.4. However, execution of most of the steps in the workflow means textually describing the actions taken in a template document. Evaluation of results is done manually by a human evaluator filling out checklists that define the criteria to be examined. For example, to compare the results of migrating Word documents to PDF, a page by page comparison is undertaken using two or more computers as showed in Figure 3.3.

3.1.3 Summary

The Dutch Testbed, which is a major input for the PLANETS Testbed, was designed to give insight into practical problems and organisational issues

Figure 3.3: Dutch Testbed laboratory



Experiment Interface - Microsoft Internet Explorer

Testbed Experiment Interface

Experiment 44 (Process 1)

Preservation approach: Migration
Record type: Document
Description

Experiment Stages

1 Define exploration area ✓
2 Prepare for experiment ✓
3 Define requirements ✓
4 Develop experiment design ✓
5 Specify resources ✓
6 Go/No go decision ✓
7 Develop experiment ✓
8 Test experiment ✓
9 Go/No go decision ✓
10 Run experiment ✓
11 ***Evaluate experiment*** ✓
12 Consider results ✓

Stage 11: Evaluate experiment
Status: Complete

Documents:
S11 - FE - Experiment-Evaluation-44_1d.doc

Change status (EO)

Experiment 44 records

OROs: WordPerfect 4.2 - 5.1	TROs: Word 2002
tr-WPdoc-UA-01 Metadata T0769.142	tr-WPdoc-UA-01-ep1-e44 Metadata tr-WPdoc-UA-01-ep1-e44.doc
tr-WPdoc-UA-02	tr-WPdoc-UA-02-ep1-e44
tr-WPdoc-UA-03	tr-WPdoc-UA-03-ep1-e44

Experiment 44

Select multiple records

Observations and notes

Relates to: Experiment 44

Subject header:

Enter text:

Figure 3.4: Software interface supporting the experiment workflow

of digital preservation, focusing on the authenticity of digital objects due to it being devised in a national archive. It features a thoroughly defined organisational structure and precisely specified workflow to provide a stable basis for the evaluation of preservation strategies. However, the process of setting up, executing and evaluating experiment is largely carried out manually.

3.2 The PLANETS Testbed

Most of the steps in the workflow of the Dutch Testbed consist merely of textual documentation. Any work in specifying, developing and setting up experiments as well as evaluating their results is done manually. For large-scale and repeated use, the situation definitely needs to be improved. The PLANETS Testbed aims at achieving this by improving the workflow support and documentation and automating the development and execution of experiments by integrating services in a distributed environment.

This section will introduce the PLANETS project and its main goals and the technical architecture that the testbed resides in and communicates with. We will then give a short overview of the core tasks identified during requirements analysis that the testbed software has to support.

3.2.1 The PLANETS Project

PLANETS – Preservation and Long-Term Access through NETworked Services – is an Integrated Project funded by the European Commission Information Science and Technologies Sixth Framework Programme (FP6 Call 5) which started June 2006 and runs for four years with a funding of about 14 million €.

PLANETS *brings together European National Libraries and Archives, leading research institutions, and technology companies to address the challenge of preserving access to digital cultural and scientific knowledge . . .*
PLANETS will deliver a sustainable framework to enable long-term preservation of digital content, increasing Europe's ability to ensure long-term access to its cultural and scientific heritage. [The06]

The consortium consists of

1. **Libraries**
    - The British Library
    - The National Library of the Netherlands

   - Austrian National Library
   - The Royal Library of Denmark
   - State and University Library, Denmark

2. **Archives**

   - The National Archives of the Netherlands
   - The National Archives of England, Wales, and the United Kingdom
   - Swiss Federal Archives

3. **Universities**

   - University at Cologne
   - University of Freiburg
   - HATII at the University of Glasgow
   - Vienna University of Technology

4. **Technology Companies**

   - Austrian Research Centers GmbH
   - IBM Netherlands
   - Microsoft Research Limited
   - Tessella Support Services Plc

The project's goals are to construct and implement effective and connected solutions in the following areas.

1. **Preservation Planning** services enable organisations to specify, evaluate, and perform preservation plans based on their specific requirements influenced both by their organisational policies and the collections they are owning.

2. **Preservation Characterisation** services extract the critical properties and characteristics of digital objects.

3. **Preservation Action** services operate on the digital objects or their environments to ensure long-term access. Examples include migration tools for digital objects or emulation environments such as the Universal Virtual Computer mentioned in Chapter 2.

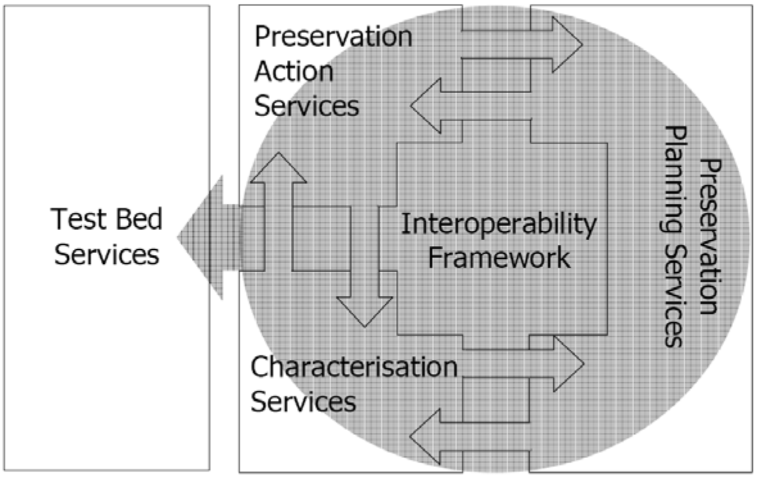

Preservation Action Services
Test Bed Services
Interoperability Framework
Preservation Planning Services
Characterisation Services

Figure 3.5: PLANETS technical sub-projects [PLA06a]

4. **Service Registries** for all of the above and an **Interoperability Framework** will integrate tools and services to provide a coherent distributed network of services.

5. The **Testbed** uses the services provided by preservation action and preservation characterisation which are discovered through the service registry to provide a consistent base of evidence to enable the repeatable, comparable, and objective evaluation of these services.

6. Finally, **Dissemination and Takeup** activities will ensure wide-spread adoption of the project results in the user community. Moreover, a dedicated goal of PLANETS is to open up a new European marketplace for digital preservation suppliers.

Figure 3.5 depicts the relationships between the technical sub-projects of PLANETS.

3.2.2 The PLANETS Distributed Service Architecture

The core concept of PLANETS is to build a distributed architecture including an interoperability framework to ensure proper coherence and integration of the complementary services. The Interoperability Framework (IF) will allow simple services to be orchestrated into complex workflows. Figure 3.6 shows the core components of the IF as described in [PLA06b].

The IF consists of several components playing together to enable smooth service integration.

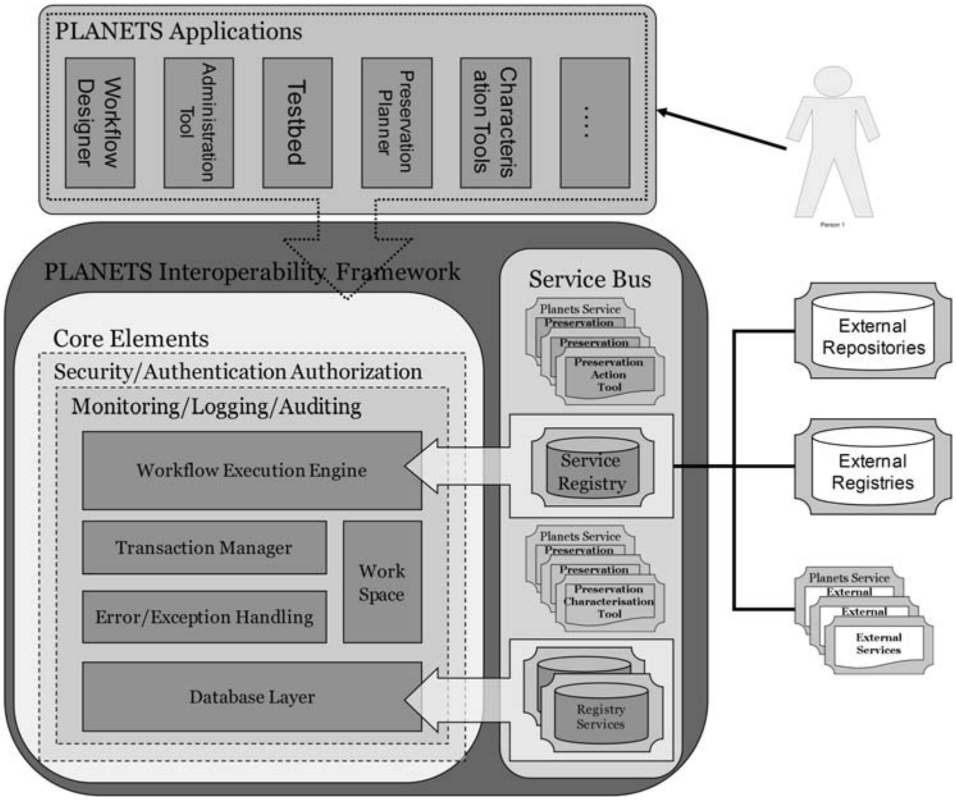


Figure 3.6: Main components of the Interoperability Framework [PLA06b]

1. A **service bus** connects tools that are encapsulated as services with the registries and external services.

2. **Basic services** include security, authorisation and authentication, monitoring and logging, auditing, error handling, and transaction management.

3. A **workflow execution engine** will be capable of executing complex workflows of preservation services defined in a dedicated tool, the **workflow designer**. With the workflow designer, a user can create complex workflows of services and input-output mappings integrating the services that are available through the Interoperability Framework. A completed and verified workflow can be deployed as a service in itself and can thus be used in experiments.

4. A **workspace** component and a **database layer** take care of persisting digital content and any metadata.

All tools and services developed during the course of PLANETS will be united and accessible through this interoperability framework. This includes preservation planning as well as preservation action and characterisation tools, and the testbed.

In contrast to the Dutch Testbed, the PLANETS Testbed sub-project will not in itself develop solutions for digital preservation such as preservation action or preservation characterisation tools. Instead, development of solutions takes place in the respective sub-projects, and the developed and tested services are deployed and accessible through the IF. The testbed system itself will be stable and integrate these services.

3.2.3 Core tasks

This section gives a short introduction to the core tasks that the testbed software has to support.

Performing experiments

This is naturally at the heart of the testbed. Users need to be able to specify in detail the kind of experiment they want to conduct, the resources they need and the services they want to use. They need notification mechanisms for events such as experiment completion, and a thoroughly specified workflow to allow for traceability of experiment definition and reproducibility of results.

Overall there are three kinds of experiments foreseen:

Formal experiment	**Informal experiment**
Each step is documented in the database; project properties, tool settings, the environment, test setup and test data are stored.	The experimenter can save test results, but they are not automatically documented.
The experimenter has to follow the complete workflow.	Several non-essential steps may be omitted.
Because everything is saved, test setups can be reused for later experiments.	Reuse of test setups is not possible.
There are no limits on the size of experiments. If the volume exceeds the threshold, approval and scheduling is necessary.	If the volume exceeds the threshold, the experimenter is notified, and the experiment is not executed.
Test data sets are automatically stored in the data registry.	Test data sets are not automatically stored in the registry; the experimenter can decide to store them.

Table 3.1: Comparing formal and informal experiments

1. **Formal experiments** follow a thoroughly defined workflow that is quite similar to that specified in the Dutch Testbed. Every step taken is monitored and logged in detail to allow traceability, reproducibility and accountability of actions and results.

2. In order to allow quick, explorative experiments without posing too much overhead on the experimenters, there will be support for **informal experiments** which have less restrictions and documentation along the process. However, there will be a limit in size imposed on informal experiments. Large-scale experiments have to follow the formal experiment process. Table 3.1 gives an overview of the commonalities and differences between formal and informal experiments.

3. Apart from these two, **preservation planning** experiments are not carried out in the testbed software, but instead in the planning tool deployed on the testbed platform. They follow their own workflow, but may access the same data repository.

Considering formal and informal experiments, there are four types of experiments from the perspective of the tools that are to be tested.

1. A **migration experiment** aims at exploring the characteristics of migration tools. To achieve this, the experimenter may optionally employ characterisation services on the original objects as well as the migrated objects. For example, when testing migration tools that convert JPEG2000 images to TIFF, characterisation tools might be used to verify the validity of the files and their correspondence to file format standards, or to extract metadata fields and verify that the migration tools have kept these metadata intact.

2. A **characterisation experiment** focuses on the abovementioned characterisation services, without considering migration or emulation.

3. **Emulation experiments** are potentially complex to realise in the distributed service-based environment formed by the PLANETS architecture. These might include traditional emulation approaches as well as modern approaches like the UVC or modular emulation described in Chapter 2. The exact nature of emulation experiments in the PLANETS Testbed is currently an open issue that will be resolved in the next stages of the project.

4. Finally, more complex configurations can be realised by conducting an **experiment based on a predefined workflow** that has been designed using the Workflow Designer described in Section 3.2.2.

Access benchmark data

A central part of experiments is defining the data set on which the experiment should execute. As a data repository is therefore an important part of the testbed system, the software needs to provide an interface to it and enforce the corresponding security constraints.

Apart from using benchmark content that is already present in the data registry, experimenters can upload their own data to test services on their own content.

Access experiment results

Users want to compare their results with previous experiment results and to search for previous experiments in order to analyse the respective data and decide if they need to conduct their own experiment. An experiment registry provides these functionalities.

Administration

In order to support the core functionality of the testbed, there is a need for **basic administration tasks** like user management, but also for specific functionality such as deleting experiment data (which should not be possible for a 'normal' user due to security reasons). Moreover, the administrator needs a means of **scheduling resource-intensive experiments**.

3.3 Summary

In this chapter, we delineated the foundations of the PLANETS Testbed. We described its predecessor, the Dutch Testbed for digital preservation, and analysed the main concepts. We then introduced the PLANETS project, its main goals and the interoperability framework which forms the glue keeping the sub-projects and their results together in a coherent and consistent way. Finally we introduced the PLANETS Testbed, its goals and the main tasks that it has to support. These tasks are specified in detail in the following chapter, where we describe the use case model for the testbed in detail.

Chapter 4

Use Case Model

4.1 Introduction

This chapter forms the core part of this work, specifying the functional model for a digital preservation testbed using the use case methodology outlined in Section 2.3.3. After an outline of the process of use case writing and the partners involved, we will introduce the main actors and roles and outline the core tasks to be supported. We will then provide the use case descriptions grouped by their functionality and include use case diagrams where appropriate. A short summary of the testbed functionality concludes this chapter.

Organisation	Abbreviation	Role
Austrian National Library	ONB	Contributor
British Library	BL	Contributor
Vienna University of Technology	TUWIEN	Use Case lead
Humanities Advanced Technology And Information Institute at the University of Glasgow	HATII	Quality Assurance
Austrian Research Centre	ARC	Quality Assurance

Table 4.1: Partners involved in the use case analysis

The concept for this functionality including proper break down of use cases and coherence checking from summary-level to subfunction was developed by TUWIEN. The project partners listed in Table 4.1 then agreed on a distribution of responsible authors, while TUWIEN reviewed the completed description and compiled the delivered use case specification. Table

4.2 lists the complete use case numbers and titles as defined for the PLANETS Testbed, together with the distribution of responsible authors for the delivered original use cases. The numbers in the use cases refer to the identifiers assigned during the process of specifying the PLANETS use cases. As usual during the process of writing and refining use cases, some of the use cases were deleted, causing some gaps in the use case numbering.

ID	Use case title	Responsible
1	Login	ONB
2	Logout	ONB
3	Perform formal experiment	TUWIEN
4	Define basic properties	ONB
5	Design experiment	TUWIEN
7	Define migration experiment	TUWIEN
8	Select a tool	TUWIEN
9	View tool details	TUWIEN
11	Configure tool	TUWIEN
12	Define test data set	BL
14	Select data from data registry	BL
15	Define characterisation experiment	TUWIEN
16	Define emulation experiment	ONB
17	Define experiment for predefined workflow	TUWIEN
18	Run experiment	TUWIEN
19	Take scientific Go-decision	ONB
21	Evaluate experiment results	ONB
22	Select viewer	ONB
23	Edit evaluation results	ONB
24	Generate report	ONB
25	Edit conclusions	ONB
26	Perform informal experiment	ONB
28	Evaluate preservation plans	TUWIEN
29	Approve experiment	TUWIEN
30	Repeat previous experiment	TUWIEN
31	View available services	ONB
34	Browse data registry	BL
35	Upload data to data registry	BL
36	Move data	BL
37	Delete data	BL
38	Copy data	BL
39	View metadata	BL
40	Edit Metadata	BL

42	Apply filter for data registry	BL
45	Manage access rights	BL
46	Browse experiment registry	BL
47	Search for experiment	BL
48	Apply filter for experiment registry	BL
49	Download experiment details	BL
50	View experiment details	BL
51	Export experiment settings	ONB
52	Import experiment settings	ONB
53a	Notify experimenter of stalled experiments	ONB
53b	Edit stalled experiment	ONB
54	View running experiments	ONB
55	Cancel running experiment	ONB
56	Delete experiment	BL
57	View experiment status	ONB
58	Take technical Go-decision	TUWIEN
59	Schedule experiment	BL

Table 4.2: Use cases and responsible authors

4.2 Actors

This section lists the roles interacting with the testbed and provides a short description for each. Please note that throughout the use case descriptions, the male form is used for the sake of readability. Of course, it refers to both genders.

1. The **Testbed Supporter** provides advice and knowledge on testbed procedures and experiments. He might want to access the experiments database etc., but is not technical but rather organisational support.

2. The **Testbed Administrator** is responsible for managing the system and solving technical issues (technical support). He is for example responsible for scheduling experiments.

3. The **Experimenter** role comprises two concepts:

   - A **Content Experimenter** owns content and wants to experiment with tools on this content. The focus lies in finding optimal solutions to preserve the content one is responsible for.

- A **Tool Experimenter** does not necessarily own content. He wants to examine the specific characteristics of different tools. This will often be done using benchmark content.

While these two roles have different perspectives, the desired functionality from the system point of view is the same. Thus they are merged into one role, the Experimenter.

4. The **Reader** role applies to users who read experiment settings and results, but have no right to conduct experiments or edit data.

5. The **Scientific Validator** is responsible for taking the scientific Go-decision. Depending on the institutional context, the same user might be assigned both the experimenter and the validator role and thus be able to execute the whole testbed workflow.

6. The **Service Provider** has one or more tools that he wants to provide as PLANETS services, but will not conduct experiments.

7. The **Testbed User** is a generalization of all users accessing the testbed software and thus using the functionality provided thereby.

Slightly set apart from these roles, we want to mention another user that is relevant in the context of the testbed. The **Preservation Planner** might be seen as a testbed user, as the preservation planning tool will be accessible through the testbed software and may be deployed on the testbed instance. However, this role is only relevant inside the preservation planning software, which is considered a black box for the testbed.

4.3 Use Cases

In the following we will provide revised versions of most of the use cases listed in Table 4.2, grouped by the tasks they are supporting. We merged several use cases and omitted some of them which are not centrally relevant in the context of this thesis to improve overview and readability and allow the reader to focus on the core tasks of the testbed. For the same reasons, we omitted the following sections of the use case descriptions:

1. Use Case ID,

2. Author,

3. Version,

4. Last updated, and

5. Issues/Comments.

The functionality of the testbed can be grouped into the following chunks of use cases:

1. **Defining Experiments** – A thorough and well-documented definition of the experiment setting, including the tool configuration, the test data set, and evaluation criteria, is of vital importance to ensure comparability and repeatable experiments. This includes migration, characterisation, and emulation experiments.

2. **Tool Configuration** – This section describes selecting and configuring services that the testbed should use for experiments.

3. **The Data Repository** – These use cases describe how the users may interact with the data repository holding the collections of digital objects that the experiments operate on. This includes means of configuring test data sets for use in an experiment.

4. **Running Experiments** – This section contains the use cases that describe the workflow of submitting experiments. Included here are means to ensure that no experiments are submitted without proper evaluation and approval, and services that allow the administrator to schedule experiments to run at a specified point in time. Stakeholders considered these features very important due to the potentially huge amount of data that will be submitted to experiments.

5. **Evaluating Experiments** – Without a proper and thorough evaluation and documentation of experiment results, the testbed would not be of any meaningful use. The use cases in this section describe how the experimenter can interact with the system to view the results of experiments, be that migration, characterisation, or emulation experiments, and record manual evaluation results as well as conclusions.

6. **Accessing the Experiment Registry** – The experiment registry holds all data about experiments both currently in definition and already completed. A fully featured user interface to access this registry is important to allow for searching and exploring previously completed experiments. This provides users the opportunity to compare their experience with the experiment results of other testbed users.

7. **Reusing Experiments** – This time-saving feature includes the option to comfortably reuse a complete experiment setting including test data and tool configuration, for example to use a new version of a previously tested migration service to compare the results with the previous experiment.

8. **Preservation Planning** – The decision support tool for preservation planning which implements the workflow described in Section 2.1.3 is considered a black box from the testbed point of view. More precisely, the workflow of the planning tool is independent of the testbed management workflow for experiments. This section contains a simple use case describing the execution of the planning tool, and for illustrative purposes also includes the main use case which describes the workflow that is executed inside the planning tool.

9. **Management and Administration** – This section finally describes general administration tasks such as handling running experiments, but also functionality such as deleting experiment data.

4.3.1 Defining experiments

The thorough and well-documented definition of experiment settings, including the tool configuration, the test data set, and evaluation criteria, is one of the primary aspects of the testbed software. The distributed, loosely-coupled web service approach taken in the PLANETS project allows discovery and dynamic integration of preservation services ranging from migration and emulation to characterisation and preservation planning. However, this also implies that the services can be discovered easily and integrated smoothly. The testbed supports the user in finding, selecting and configuring appropriate services as needed. The corresponding use cases are described in this section.

Figure 4.1 shows the main summary-level use case of the system, **Perform Formal Experiment**, which integrates a number of use cases at the user-goal level to provide the desired functionality. One of the main components of this task consists of **designing an experiment**, i.e. the configuration of test data and tools to be tested. Figure 4.2 shows the use case diagram for designing an experiment; the use case itself and those it references are provided in the next section.

The use case diagrams are intended to provide an overview on the hierarchy of use cases; they are not meant to be complete. Specifically, they do

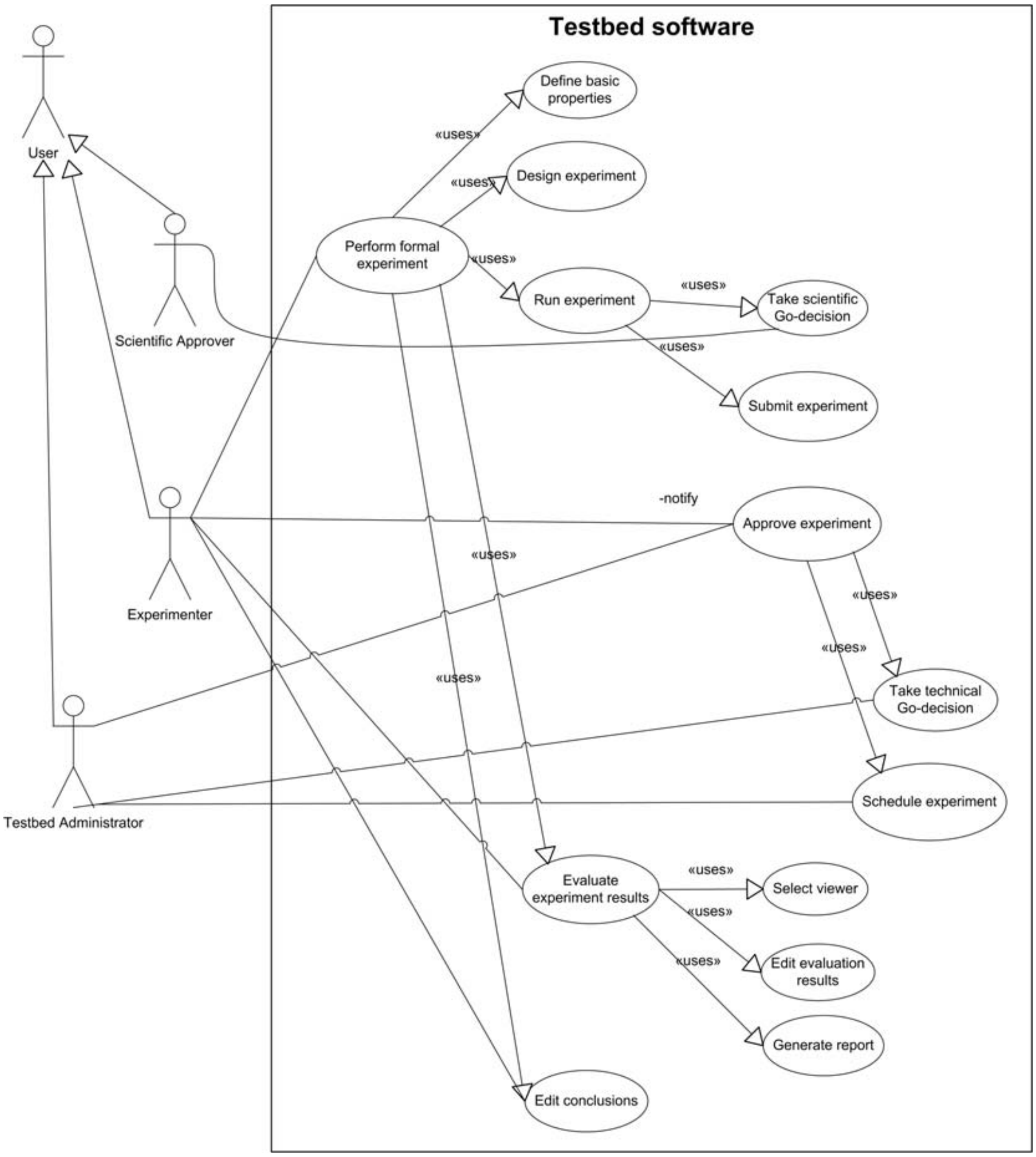


Figure 4.1: Perform formal experiment

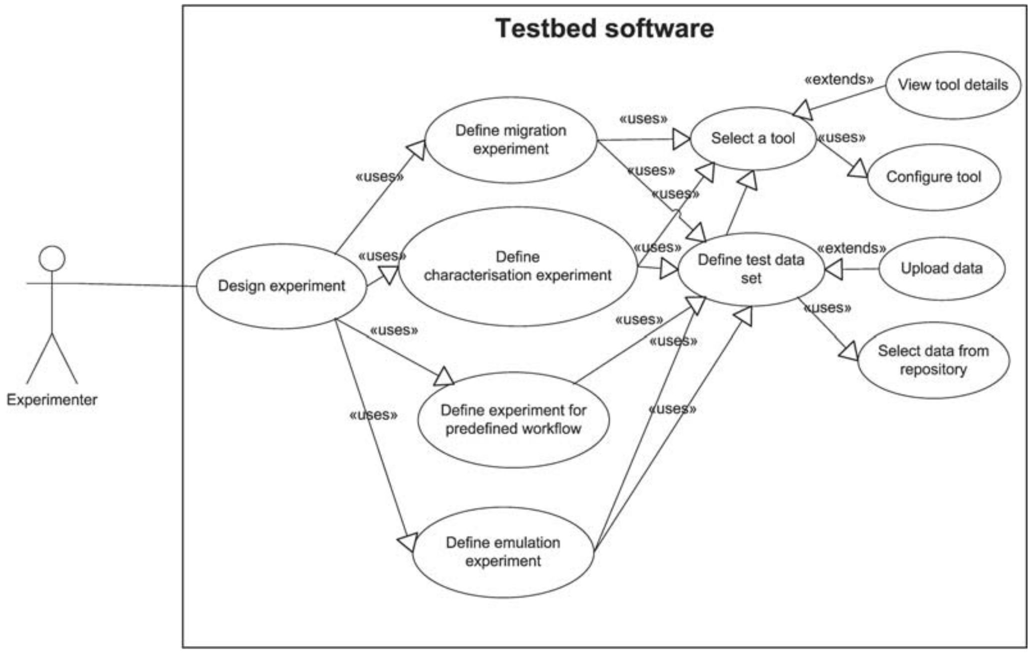

Testbed software
Experimenter
Design experiment
Define migration experiment
Define characterisation experiment
Define experiment for predefined workflow
Define emulation experiment
Select a tool
Define test data set
View tool details
Configure tool
Upload data
Select data from repository
«uses»
«extends»

Figure 4.2: Design experiment

not show all use cases that are included down to the lowest level. For example, **Design experiment** consists of a number of steps that are modelled in use cases, but are not shown in the diagram for reasons of clarity. Please refer to the textual descriptions for details on these inclusions. Moreover, the communication links between lower-level use cases and the experimenter have been omitted for reasons of readability.

Perform formal experiment

Scope	Testbed Software
Level	Summary
Actors	Experimenter
Preconditions	The experimenter is logged in.
Trigger	The experimenter selects the according function in the main menu.
Success guarantees	A formal experiment is defined and performed, all relevant data, including test data and outputs, are stored and accessible.

Main Success Scenario	
1	The experimenter defines the basic properties of the experiment.
2	The experimenter designs the experiment.
3	The experimenter runs the experiment.
4	The experimenter evaluates the experiment results.
5	The experimenter edits the conclusions.
Exceptions	
-	none

Define basic properties

Scope	Testbed Software
Level	Subfunction
Actors	Experimenter
Preconditions	The experimenter is performing a formal experiment.
Trigger	The experimenter selects the according function in the menu.
Success guarantees	The basic properties for an experiment are defined.
Main Success Scenario	
1	The system presents the user with a form requiring scope, purpose, and other metadata for the experiment.
2	The user enters the information and submits the data.
3	The system verifies the data and saves the basic properties.
Exceptions	
2	The user cancels: The system returns to the Perform formal experiment interface.
3	Some properties are not valid: The system notifies the user, continue with 1.

Design experiment

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	The experimenter is working on a formal experiment of which at least the basic properties have been defined.
Trigger	The experimenter has finished defining basic properties.
Success guarantees	The experiment design is completed and stored; the experimenter can submit it for approval.

Main Success Scenario

1 The experimenter decides on a type of experiment. This means one of

1. Migration,
2. Characterisation,
3. Emulation, or
4. A predefined workflow modelled in the workflow designer of the Interoperability Framework.

2 The system records the decision and presents an according screen.

3 The experimenter defines the experiment. Depending on the type of experiment decided in the previous step, this means either

1. define a migration experiment,
2. define a characterisation experiment,
3. define an emulation experiment, or
4. define an experiment for a predefined workflow.

Exceptions

1 The experimenter cancels: The system returns to the previous screen, nothing is changed in the data.

As described in the use case above and depicted in Figure 4.2, there are four types of experiments.

1. A **migration experiment** aims at exploring the characteristics of migration tools. To achieve this, the experimenter may optionally employ characterisation services on the original objects as well as the migrated objects.

2. A **characterisation experiment** focuses on characterisation services, without considering migration or emulation.

3. **Emulation experiments** are potentially complex to realise in the distributed service-based environment formed by the PLANETS architecture. These might include traditional emulation approaches as well as modern and emerging approaches like the UVC or modular emulation described in Chapter 2. Due to their complexity, work on integrating them in the testbed is still in progress.

4. Finally, more complex configurations can be realised by conducting an **experiment based on a predefined workflow** that has been designed using the Workflow Designer mentioned in Section 3.2.2.

We describe the process of designing these types of experiments in the following in the order they are mentioned above. The section is concluded by the use case describing an informal experiment.

Define migration experiment

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	The experimenter is designing an experiment that has thus at least completed the step Define basic properties.
Trigger	The experimenter selects a migration experiment in Design an experiment.
Success guarantees	A migration experiment is defined and all relevant configuration data stored.
Main Success Scenario	
1	The system presents an overview interface showing the current configuration of tools. This means basically three blocks: 1. Characterisation before migration 2. Migration tools 3. Characterisation after migration The reasoning is that migration experiments usually need to rely on characterisation to achieve comparable results in an objective way. The experimenter needs characterisation informing him about differences that he achieved by performing the migration; hence the characterisation steps before and after migrating the objects. Characterisation tools here always work either on the input data or the output of the migration tools used, similar to the concept of ‘original record object’ and ‘transformed record object’ in the Dutch Testbed.

	We envision an interface showing three boxes: characterisation on top, migration in the middle, and another characterisation at the bottom. The advantage of this compared to other more generic settings with more freedom is that the structure is clearer and easier to implement. More complex workflows can be designed with the workflow designer and used with the corresponding use case.
2	The experimenter iteratively selects a tool for each of these blocks. At least one migration tool is required, defining characterisation tools is optional. Alternatively, he can also remove a tool and its associated configuration from the list by selecting it and selecting an according function. The system should ask for confirmation before removing the selected tool from the configuration.
3	When finished, the experimenter submits the data.
4	The system verifies that at least one tool was selected, and saves the configuration.
5	The experimenter defines the test data set.
Exceptions	
4	No migration tool was selected: The system notifies the experimenter, who can add a tool. Continue with 1.

Define characterisation experiment

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	The experimenter is designing an experiment that has thus at least completed the step Define basic properties.
Trigger	The experimenter selects a characterisation experiment in Design an experiment.
Success guarantees	
Main Success Scenario	
1	The system presents an overview interface showing the current configuration of tools. This would look quite like the interface for defining a migration experiment, but with only one block: Characterisation.
2	The experimenter iteratively selects a tool. At least one tool is required. Alternatively, he can also remove a tool and its associated configuration from the list by selecting it and selecting an according function. The system should ask for confirmation before removing the selected tool from the configuration. (Note that the user can select the same tool more than once and use varying configuration settings. The use case Select tool includes tool configuration.)

3	The system verifies that at least one tool was selected, and saves the configuration.
4	The experimenter defines the test data set.
Exceptions	
3	No tool was selected: The system notifies the experimenter, who can add a tool. Continue with 1.

Define emulation experiment

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	The user is designing an experiment that has thus at least completed the step Define basic properties.
Trigger	The experimenter selects an emulation experiment in Design an experiment.
Success guarantees	The emulation experiment is defined.
Main Success Scenario	
1	1. The system presents an interface showing the components for the definition of the experiment: 1. Settings for the Reference Environment, i.e. the environment to be emulated: (a) Hardware, (b) Operating System, and (c) Tools to be used for rendering the data. 2. Tools for characterisation after emulation 3. Tools for emulation or virtualisation service, along with their settings.
2	The experimenter iteratively selects a tool or setting for one of these blocks. Defining characterisation tools is optional.
3	The user defines the test data set.
Exceptions	
-	none

Define experiment for a predefined workflow

Scope	Testbed Software
Level	User Goal

Actors	Experimenter
Preconditions	The experimenter is designing an experiment that has thus at least completed the step Define basic properties.
Trigger	The experimenter selects an experiment for a predefined workflow in Design an experiment.
Success guarantees	An experiment for a predefined workflow is defined and all relevant configuration data stored.
Main Success Scenario	
1	The system presents an overview interface showing the current configuration. This would look quite like the interface for defining a migration experiment, but with only one entry, the workflow that has been loaded.
2	The experimenter selects a predefined workflow (by using the Select Tool use case).
3	The system verifies that a valid workflow was selected and all necessary parameters have been configured, and saves the configuration.
4	The experimenter defines the test data set.
Exceptions	
3	No workflow has been selected: System notifies user, who can add a workflow. Continue with 1.

Perform informal experiment

Scope	Testbed Software
Level	User Goal/Summary
Actors	Experimenter
Preconditions	The user is logged in.
Trigger	The experimenter selects the function in the main menu.
Success guarantees	An informal experiment is successfully performed.
Main Success Scenario	
1	The experimenter designs the experiment.
2	The experimenter runs the experiment.
3	The experimenter evaluates experiment results as needed.
Exceptions	
-	none

4.3.2 Tool Configuration

Tool Configuration in the PLANETS Testbed consists of two steps.

1. First the experimenter will **discover and select a service** through semi-automated support from the software. This means e.g. that the

service registry will be consulted to provide a list of services conforming to the users needs.

2. After selecting a specific service, the experimenter will often need to **configure the service**, which means setting parameters and possibly defining the kind of output that is desired.

These steps apply to all kinds of experiments, including predefined workflows. The latter are deployed as PLANETS services like normal tools, so that they can be configured similar to the former.

Select a tool

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	Experimenter is defining a migration, characterisation or emulation experiment.
Trigger	Experimenter selects 'Add tool' to add a tool at a specific location.
Success guarantees	A tool is selected, configured and added to the corresponding list; the configuration data are stored.
Main Success Scenario	
1	The system loads and presents a list of tools that are applicable for the current context and accessible according to the access control restrictions for users, such as migration tools.
2	The experimenter can apply filtering mechanisms to this list and sort it according to different criteria.
3	The experimenter can view tool details for each tool by selecting it.
4	The experimenter selects one tool.
5	The system adds this tool to the experiment and presents an interface for setting parameters for this tool.
6	The user configures the tool and submits the data.
7	The system saves the configuration and returns to the screen from where tool definition was initiated; the tool is inserted at the correct location, i.e. in the right category where the experimenter selected to add a tool.
Exceptions	
1	The system cannot find applicable tools: It notifies the experimenter, who can return to the previous screen.
4	The experimenter does not select a tool, but cancels the operation: No tool definition is added, the system returns to the point from where tool selection was initiated.

6	The experimenter cancels: No tool definition is added, the system returns to the point from where tool selection was initiated.

View tool details

Scope	Testbed Software
Level	Sub-function
Actors	Experimenter; Reader, Testbed support, Testbed administrator
Preconditions	The user is viewing a list of available tools, e.g. while selecting a tool.
Trigger	The user selects one of the tools to view detailed information.
Success guarantees	The system displays all available information about the tool.
Main Success Scenario	
1	The user selects a tool for which he wants to view detailed information. The system shows detail information on the selected tool. This includes 1. Name, description, version information; 2. Provider, characteristic service parameters, ratings, etc.; 3. Necessary and optional configuration parameters; 4. Input data; and 5. Description of output.
2	The user views the data. When finished, he selects an exit button or link.
3	The system returns to the list of tools.
Exceptions	
-	none

Configure tool

Scope	Testbed Software
Level	Sub-function
Actors	Experimenter
Preconditions	The experimenter is selecting a tool.
Trigger	The experimenter selects a tool from the list of available tools.

Success guarantees	The tool is configured appropriately, and all relevant configuration data are stored.
Main Success Scenario	
1	The system presents a sort of key-value table of parameters to set, indicating which are mandatory and which are optional. Where parameters are to come from a set of possible values, the system should present these as a list to select from. If available, the system also presents a list of predefined default configurations.
2	The user sets these parameters. When finished, he submits the data. Alternatively, he can select one of the predefined default configurations.
3	The system verifies that all required data is present and that the input is correct, and saves the configuration.
Exceptions	
3	Required configuration parameters are missing: The system notifies the user, providing information about which parameters are missing. Continue with 1.
3	Configuration parameters are incorrect, i.e. do not conform to a specified format (like a String instead of a number) or exceed certain boundaries: System notifies user with detailed information about the error. Continue with 1.

4.3.3 The Data Repository

An indispensable part of experiments is defining the data set on which the experiment should execute. As a data repository is therefore an important part of the testbed system, the software needs to provide an interface to it and enforce the corresponding security constraints.

The functionality needed for accessing the data repository can be divided into two blocks:

1. **Defining the test data set** by selecting appropriate digital objects stored in the data repository, and

2. **Browsing and managing the data repository** to search for and manage the objects contained therein, including the access rights.

Define test data set

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	The user is conducting an experiment.

Trigger	The user has successfully selected and configured a tool.
Success guarantees	The test data configuration is stored.
Main Success Scenario	
1	The system presents the current test data set, which at the start is empty.
2	The user iteratively either 1. Uploads data to the data registry or 2. Selects data from the data registry.
3	The user selects an exit button or link to return to the experiment with the currently chosen files.
4	The system stores the changes and makes the chosen files accessible to the experiment.
Exceptions	
-	none

Select data from data registry

Scope	Testbed Software
Level	Sub-function
Actors	Experimenter
Preconditions	The user is defining the test data set for an experiment.
Trigger	The user activates the function for selecting data from the data registry.
Success guarantees	Some files are chosen for use in an experiment.
Main Success Scenario	
1	The system provides a hierarchical list of files and folders stored in the data registry (cf. Windows Explorer, etc.). The files and folders visible will be according to the access control (public, planets, private) of those files and folders; inaccessible objects will not appear at all.
2	The user performs any combination of the following to refine the display of what he sees: 1. Sort files according to name, type, size, date, owner 2. Filter/search files according to name, type, size, date, owner.
3	The user selects which files he wants to use (e.g. through a checkbox or a 'select all' button).

4	The system adds the selected files to the test data set configuration and returns to Define test data set.
Exceptions	
-	none

Browse data registry

Scope	Testbed Software
Level	User Goal
Actors	Testbed User
Preconditions	The user is logged in.
Trigger	The user selects the function in the main menu.
Success guarantees	The user was able to browse the data registry.

Main Success Scenario

1 The system provides a hierarchical list of files and folders stored in the data registry (cf. Windows Explorer, etc.). The files and folders visible will be according to the access control (public, planets, private) of those files and folders; inaccessible objects will not appear at all.

2 The user can sort, filter, and search files according to

- name,
- type,
- size,
- date, and
- owner.

3 For any file uploaded by the users own institution, the user can

1. View and edit metadata
2. Download file
3. Delete file
4. Copy or move file to another location within the data registry
5. Set access rights to one of the three levels of visibility of content that are applicable in the data registry:

1. **Public** data is visible to all users of the testbed.
2. **Planets** data are visible to testbed users who are member of the PLANETS project consortium.
3. **Private** data are only visible to the owning organisation, i.e. users belonging to the same organisation as the one who uploaded the data.

The user can also upload files to the data repository.

4 For any file that a user can access that was uploaded by another institution the user can perform any of the following operations:

1. View metadata
2. Download

5 The user can also select multiple files at once (e.g. through a checkbox or a ”select all” button) to perform the same set of actions to all selected files.

Exceptions

- none

As the user interface for the data registry will be a standard content repository interface, we do not describe the functionality of handling data and files in detail here. For a complete description, please refer to [BSN+06], which provides use cases for all the steps mentioned in step 3 of the use case **Browse data registry** described above.

4.3.4 Running Experiments

After an experiment has been defined, the testbed software needs to provide appropriate functionality to ensure timely execution, but also a proper process for verifying that the experiment is feasibly designed and will not e.g. block the system for other users by imposing too much workload. In conjunction to that, the testbed administrator needs a means for scheduling resource-intensive experiments.

There are two decisions involved in performing a formal experiment, as described in the corresponding use case and shown in Figure 4.1:

- The **scientific Go-decision** looks at the value of the experiment for the performing organisation and judges its appropriateness and usefulness. This decision may be taken only by a dedicated role. However, especially smaller organisations may not have this role, so it should be possible to assign this role to the same user as the experimenter role.

- If the experiment has been approved by the Scientific Validator, the system calculates its size. This calculation will for the moment be a rather rough estimate based on the number of files and their size and the number of services involved. If the experiment is below a certain threshold which depends on the hardware configuration and is set by the administrator, the experiment can be executed right away. If not, however, it has to be approved. This approval process is described below in the use case **Approve experiment** and includes two parts:

  1. The **technical Go-decision** taken by the administrator judges the feasibility and the workload the experiment will impose on the system.
  2. If is is approved, the administrator will **schedule the experiment** as described in the corresponding use case.

Run experiment

Scope	Testbed Software
Level	User Goal / Summary
Actors	Experimenter, Scientific Validator, Administrator
Preconditions	The experimenter is working on a project that is fully defined and ready to perform.
Trigger	The experimenter completes the preceding step, Define an experiment.
Success guarantees	The experiment is successfully executed, and the system notifies the user upon completion. All relevant input and output data are stored.
Main Success Scenario	
1	The scientific validator takes the scientific Go-decision.
2	The system stores the decision and presents an overview screen summarising the settings of the experiment.
3	The experimenter may enter a message to be sent to the testbed administrator. He then submits the experiment for execution.
4	The system verifies that the volume of the experiment is below the threshold for approval and scheduling.
5	The system runs the experiment. While the experiment is running, the experimenter can view the experiment status.
6	Upon completion, the system notifies the user by email.

Exceptions

1 The scientific validator takes the No-Go decision: It is not possible to submit the experiment. As with the Go-decision, the reasoning for this decision is stored in the registry. The experimenter can still revise this decision by taking the go-decision again.

3 The experimenter decides to cancel: The experiment is not submitted, but the scientific Go-decision is still valid and stored. The experimenter can submit the experiment later on.

4 The volume of the experiment is above the threshold, thus the experiment has to be approved and scheduled: The system starts the use case Approve experiment. Upon successful completion of this use case, continue with 4.

Take scientific Go-decision

Scope	Testbed Software
Level	User Goal
Actors	Scientific Validator
Preconditions	The testbed software is up and running, all relevant services are available.
Trigger	The user receives notification about a pending experiment waiting approval.
Success guarantees	The experiment is marked as executable.

Main Success Scenario

1 The user logs in to the testbed using the link provided in the notification email.

2 The system verifies the identifier contained in the link and displays a form for taking the scientific Go-decision.

3 The user enters motivation for his decision in the form, saves the motivation and submits the data.

4 The system verifies that the user took a positive decision, marks the experiment as executable and sends notification to the experimenter.

Exceptions

1 Login failed: The user cannot take the scientific decision.

2 The user did not log in with the link provided in the email or the identifier cannot be found: The user has to search for the experiment. After he has found it, continue with 2.

2 The user has no permission to take the scientific Go-decision: The system presents a corresponding message, the user cannot take the scientific Go-decision.

3 The user cancels: The system does not change any data and does not mark the experiment as executable. It returns to the experiment page.

4	The user does not approve the experiment, i.e. does take the No-Go decision: The system stores this and returns to the main menu.

Approve experiment

Scope	Testbed Software
Level	User Goal
Actors	Experimenter, Administrator
Preconditions	An experiment has been defined and submitted for execution.
Trigger	A submitted experiment exceeds the volume threshold and thus cannot be executed without approval.
Success guarantees	The experiment is approved, scheduled and run; the experimenter is notified about progress and completion.
Main Success Scenario	
1	The system notifies the experimenter that the experiment has to be approved.
2	The system notifies the administrator about the pending approval and scheduling by email, providing an overall summary of experiment type, tools, and test data of the experiment.
3	The administrator approves the experiment (i.e. takes the technical Go-decision).
4	The administrator schedules the experiment.
5	The system notifies the experimenter about the decision and the scheduled execution time.
6	At the specified point of time, the system runs the experiment.
7	Upon completion, the system notifies the experimenter.
Exceptions	
3	The administrator decides not to approve the experiment: The system notifies the experimenter that the experiment has been rejected. The decision is stored along with the experiment, the experimenter can redefine settings, test data, etc., and later-on resubmit the experiment again.

Take Technical Go-decision

Scope	Testbed Software
Level	User Goal
Actors	Administrator

Preconditions	A submitted experiment exceeds the volume threshold and thus has to be approved and scheduled. The system thus has notified the administrator about pending approval and scheduling. The administrator is logged in.
Trigger	The administrator selects an according function in the testbed software. If the notification email sent by the system includes a direct link, the administrator can activate this link, enter his password, and thus directly enter this use case.
Success guarantees	The decision to continue is recorded and the experiment can be scheduled.
Main Success Scenario	
1	The system presents an interface with detailed information about the pending experiment, including the number of files and their size, the tools to be executed, etc., along with the decision options: 1. Go 2. No Go
2	The administrator selects one of these, enters the reasoning and actions to be taken and selects ’Take decision’.
3	The system verifies that the administrator has chosen the *Go* option and stores the decision and the reasoning. It also notifies the experiment owner about the decision.
Exceptions	
3	Reasoning is missing: The system notifies the administrator, who can provide the missing information (continue with 1).
3	Decision is *No Go*: The system notifies the experiment owner that the experiment was rejected, and does not execute the experiment.

Schedule experiment

Scope	Testbed Software
Level	Sub-Function
Actors	Administrator
Preconditions	The administrator is approving an experiment and has taken the technical Go-decision .
Trigger	The administrator finishes the step Take technical Go-decision while approving the experiment.

Success guarantees	The experiment will run at the specified point in time.
Main Success Scenario	
1	The system presents the user a calendar-style display to show system workload.
2	The administrator enters a time to run the experiment.
3	The system schedules the experiment to run at the specified time and sends a notification to the owners of the experiment to tell them when it will run.
Exceptions	
-	none

4.3.5 Evaluating Experiments

The last stage of an experiment consists of evaluating the outcome, for which the testbed should provide a range of options. These include

- rendering migrated files in an appropriate viewer,
- allowing the user to download migrated files,
- running an emulator,
- display characterisation results, and
- generating reports.

As we laid out in the previous chapter, there has not been any automated evaluation of, for instance, migrated documents, up to now. PLANETS aims at improving this situation by at least partially automating the evaluation process, particularly in the context of automated mass-migration. Characterisation services will be able to substantially relieve the users of tedious tasks such as comparing fundamental characteristics of migrated objects such as the dimensions of images or the number of pages in documents. As further progress will be made, also more complex properties will be covered.

The following use cases provide an overview of the options an experimenter has for evaluating the outcome of experiments and record his conclusions.

Evaluate experiment results

Scope	Testbed Software
Level	User Goal
Actors	Experimenter, Reader

Preconditions	The user is logged in to the testbed, an experiment is selected, input and output data are available.
Trigger	A report for the experiment is generated.
Success guarantees	The user was able to view all data relevant to evaluation. This use case does not change any data.
Main Success Scenario	
1	The system presents an overview interface.
2	The user iteratively selects one of the following options: 1. Select viewer and platform for input data 2. Select viewer and platform for output data 3. Display characterisation results for input data 4. Display characterisation results for output data 5. Edit evaluation results 6. Generate report on a variety of data such as, e.g., performance values collected during experiment execution or the output of characterisation services.
Exceptions	
-	none

Select viewer

Scope	Testbed Software
Level	Subfunction
Actors	Experimenter, Reader
Preconditions	The user is evaluating an experiment.
Trigger	The user selects the appropriate option in the overview interface while evaluating an experiment.
Success guarantees	
Main Success Scenario	
1	The system presents a list of applicable viewers along with the platforms for which they are available.
2	The user selects one viewer from the list.
3	The system opens the chosen viewer in a new window.
Exceptions	
4	The user cancels: The system returns to the list of available viewers, continue with 1.

Edit evaluation results

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	The user is evaluating experiment results.
Trigger	The user selects the appropriate function while evaluating experiment results.
Success guarantees	Edited evaluation results are stored.
Main Success Scenario	
1	The system displays a form with the evaluation form allowing the user to edit results.
2	The user edits evaluation results as needed and submits the data.
3	The system verifies the consistency of data and stores the results.
Exceptions	
2	The user cancels: The system does not save the data and returns to the previous screen.
3	Required data are missing or data are inconsistent: The system notifies the user, continue with 1.

Generate report

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	Evaluation results for an experiment are available, the experimenter is evaluating experiment results.
Trigger	The user activates the according function in the menu.
Success guarantees	A report is generated and displayed to the user.
Main Success Scenario	
1	The system presents a choice of settings.
2	The user selects one of the available report types.
3	The system generates the selected report and presents it to the user.
Exceptions	
2	The user cancels: Return to experiment evaluation.

Edit conclusions

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	An experiment has run and the results have been evaluated.

Trigger	The user activates the according function in the menu or has finished evaluating experiment results.
Success guarantees	Conclusions for the experiment are stored.
Main Success Scenario	
1	The system presents a form allowing the user to edit conclusions.
2	The user edits the conclusions and the recommendations in the form. When finished, the user submits the data.
3	The system verifies the data and stores the results.
Exceptions	
3	Required data are missing or data are inconsistent: The system notifies the user, continue with 1.

4.3.6 Accessing the Experiment Registry

The experiment registry gives users access to previously performed experiments, allowing them to

- search for previous experiments in order to analyse the respective data and decide if they need to conduct their own experiment, and
- download experiments to analyse them offline in detail or save them for documentation purposes.

Another important aspect is the feature of reusing experiments, e.g. reusing the configuration settings of a previous experiment to run the same tools on different test data. This functionality is described in Section 4.3.7.

The experiment registry will be accessible through a standard repository interface similar to the interface to the data registry described in Section 4.3.3.

Browse experiment registry

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	Results from an experiment are available. The user is logged in.
Trigger	The user opens the experiment registry by activating the according function in the main menu.
Success guarantees	The user can locate the details of a previous experiment.
Main Success Scenario	
1	The system displays a repository interface to the experiment registry.

2 The user browses through a list of previous experiments. He can optionally

1. Sort and filter the list according to
    (a) Experiment name
    (b) Experiment type
    (c) Experiment outcome
    (d) Experiment date
    (e) Owning institution
    (f) Arbitrary descriptive metadata
2. Search for an experiment.

In addition to these fields, the system shall also present a status information, particularly for the experiments that have been designed, but are not yet completed. Possible states of a designed experiment are

1. waits for Go-decision,
2. waits for submission,
3. waits for scheduling,
4. is running,
5. has successfully terminated,
6. has been cancelled (e.g. due to memory, cpu, or time problems)

3 When the user selects an experiment, the system displays detailed information on the selected experiment, including the type of experiment, the tools used, the evaluation results, and any other metadata associated with the experiment.

4 The user experimenter may select an option to reuse an experiment setup as a basis for a new experiment.

Exceptions

- none

Search for an experiment

Scope	Testbed Software
Level	Sub-function

Actors	Experimenter, Reader
Preconditions	The user is browsing the experiment registry.
Trigger	The user selects the according function in the menu.
Success guarantees	The user can locate a previous experiment according to defined search criteria.
Main Success Scenario	
1	The system presents a search mask.
2	The user enters a search term (optionally including wildcards) and optionally selects fields to search. These include: • Experiment name, type, and date • Description of experiment outcome and conclusions, • Owning institution • Arbitrary descriptive metadata
3	The system returns a lists of hits.
Exceptions	
2	The user cancels: The system returns to the previous screen.

Download experiment

Scope	Testbed Software
Level	Sub-function
Actors	Experimenter, Reader
Preconditions	The user has selected a previously executed experiment from the experiments registry and is viewing the details of the experiment.
Trigger	The user activates the function to download the complete experiment.
Success guarantees	The user has downloaded the complete experiment package.
Main Success Scenario	
1	The user is provided with a standard save box, and is able to download and save the complete experiment. These might include the type of experiment, the tools and the data set used, the evaluation results, files delivered as output from an experiment and any other metadata associated with the experiment.
2	The system verifies that the size of the experiment makes downloading feasible and that the user is allowed to download the experiment data.
3	All details are downloaded to the user´s machine.

Exceptions	
2	The size of the experiment is too large for downloading: The system notifies the user, the experiment is not downloaded. The user can download all information about the experiment apart from input and output data.
2	The user is not allowed to download the test data, because it is not owned by his organisation and is not freely available: The system notifies the user, the experiment is not downloaded. The user can download all information about the experiment apart from input and output data.

4.3.7 Reusing Experiments

The option to reuse previous experiments is considered to be very helpful by most stakeholders. It will save experimenters from the tedious task of reproducing experiment settings, both on the same system and on another instance of the testbed running on a different machine.

This section thus describes two distinct, but related features:

1. **Repeating a previous experiment** means directly reusing it as basis for a new one to be defined and executed on the same machine, probably changing either the test data or the tool configuration.

2. In contrast to this, **export of experiments** is primarily intended to allow external documentation and storage of experiments. Additionally, it is possible to **import experiment settings** into a potentially distinct testbed installation to reproduce an experiment on another system. On importing these settings, the system will verify the applicability, considering availability of both test data and the services that the experiment is employing.

Repeat previous experiment

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	The experiment database contains completed experiments. The testbed is running and the experimenter is logged in.
Trigger	An experimenter browsing the registry selects to reuse an experiment setup as a basis for a new experiment.

Success guarantees	A previously completed experiment is taken as the basis of a new experiment. Settings and data are reused and a new experiment is defined.
Main Success Scenario	
1	The system presents a list of options for parts of experiment data that can be reused: • Basic properties • Test data set • Type of experiment • Tools and configuration Results of experiment cannot be copied.
2	The user selects which parts to reuse.
3	The system copies the data and sets a reference to the experiments of which the data are copied. Continue with step 1 of a new experiment. The selected data are available and may be changed further on without affecting the experiment that has been reused as basis.
Exceptions	
-	none

Export experiment settings

Scope	Testbed Software
Level	Sub-function
Actors	Experimenter, Reader
Preconditions	The user has selected a previously executed experiment from the experiments registry and is viewing the details of the experiment.
Trigger	The user activates the function for exporting experiment settings.
Success guarantees	The complete experiment settings are exported and saved by the user on his local system.
Main Success Scenario	
1	The system presents a dialog for saving the experiment to the user's local system.
2	The user selects a path and filename and confirms.
3	The system exports all experiment settings to a file and transmits the file to the user.
4	The file is stored at the user's system.

Exceptions	
-	none

Import experiment settings

Scope	Testbed Software
Level	User Goal
Actors	Experimenter
Preconditions	The user is logged on.
Trigger	The user selects the according function in the main menu.
Success guarantees	

Main Success Scenario	
1	The system asks for the file containing the experiment settings to import.
2	The user uploads a file containing experiment settings to the testbed or selects a file from the data registry. This file needs to conform to the format in which experiment settings are exported.
3	The system checks the validity of the uploaded file. It imports the settings and displays them.
4	The system allows the user to edit the basic properties of the experiment.
5	The user views the details of the experiment, might edit the basic properties, and confirms that he wants to import the settings.
6	The system imports the settings to the current experiment and stores them.
Exceptions	
3	The file is not valid: The system notifies the user, continue with 2.
3	Some of the imported settings are not applicable to the testbed instance: The system notifies the user, continue with the main success scenario.

4.3.8 Preservation Planning

To provide the context for understanding the relationship between the preservation planning workflow and the testbed, we include here, apart from the use case **Evaluate preservation plans** defined in the testbed, the use case description of the central use case of the decision support tool for preservation planning, which runs as a separate black-box application inside the testbed and shares some data with the testbed system. Figure 4.3 provides an overview of the context of this use case and the lower-level use cases that are called by it inside the planning tool.

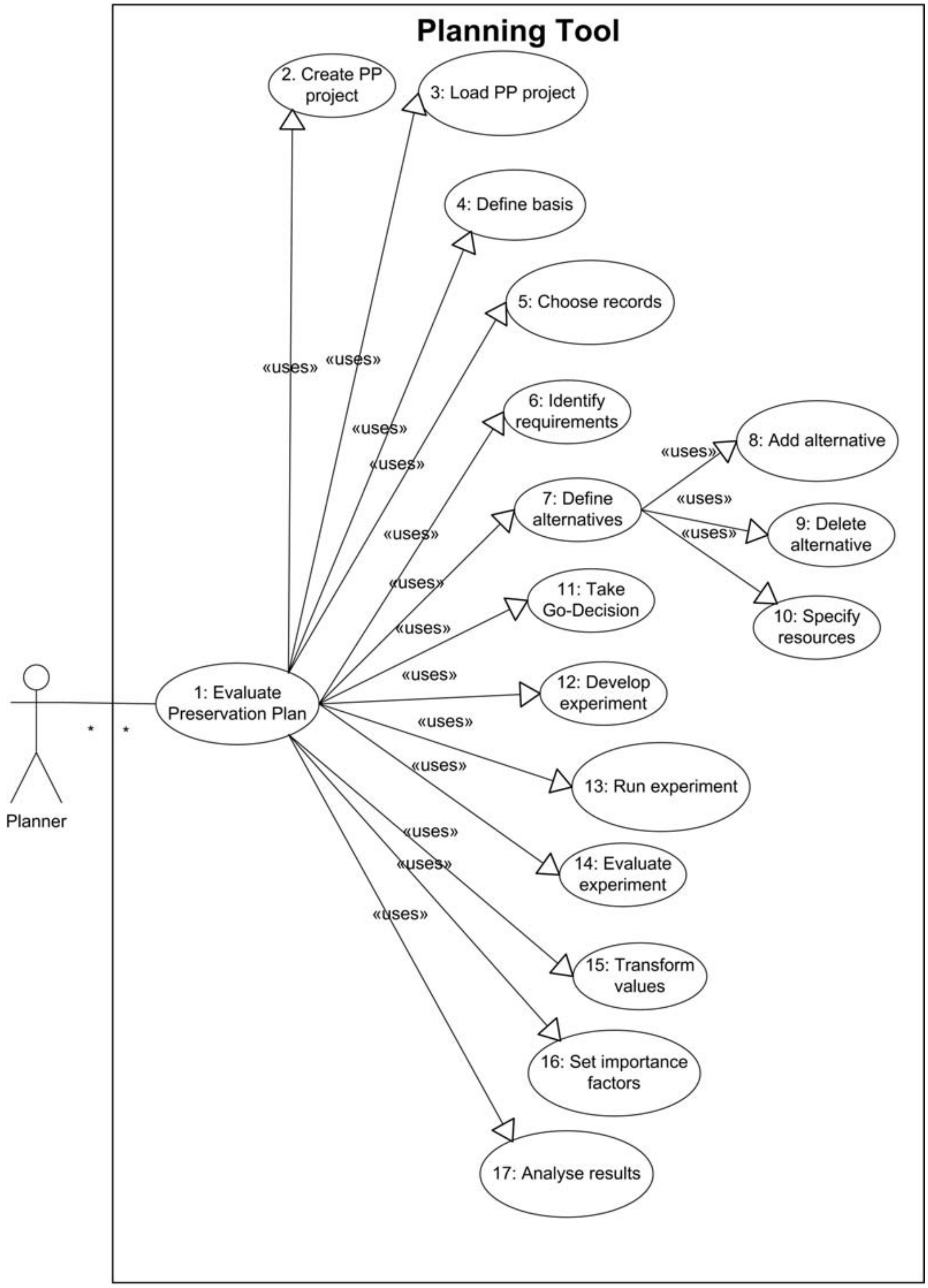


Figure 4.3: Evaluate preservation plan

Evaluate preservation plans

Scope	Testbed Software
Level	Summary
Actors	Experimenter
Preconditions	Experimenter is logged in.
Trigger	The experimenter selects the according function in the main menu of the testbed software.
Success guarantees	The planning tool is executed and preservation plans are evaluated inside the planning tool. Relevant data are stored in the testbed data registry.
Main Success Scenario	
1	The system starts the 'Planning Tool'.
2	The user follows the workflow of the planning tool, which is described in Section 2.1.3 and the following use case and depicted in Figure 2.1.
Exceptions	
-	none

Planning Tool: Evaluate preservation plan

Scope	Testbed Software
Level	Summary
Actors	Preservation Planner
Preconditions	
Trigger	The planner creates a preservation planning project or loads an existing one.
Success guarantees	The preservation planning project is evaluated, the results are stored and known to the user.
Main Success Scenario	
1	The planner defines the basis of the preservation plan.
2	The planner chooses records.
3	The planner identifies the requirements by defining the objective tree and assigning measurement units.
4	The planner defines preservation alternatives and their resources.
5	The planner takes the Go-decision.
6	The planner develops the experiment.
7	The planner runs the experiment.
8	The planner evaluates the experiment.
9	The planner transforms values.
10	The planner sets importance factors.
11	The planner analyses the results.

Exceptions

5 The planner decides not to continue (No-Go decision): The decision is recorded and stored together with the reasoning, execution of the following steps of the workflow is not possible. The planner can later resume the evaluation of the plan, e.g. if assumptions, preferences or environmental conditions have changed, by taking the Go-Decision.

4.3.9 Management and Administration

In order to support the core functionality of the testbed, there is a need for **basic administration tasks** like user management, but also for specific functionality such as deleting experiment data. It seems sensible to restrict this deletion to the administrator due to security reasons. Moreover, the administrator needs a means of **handling running experiments**, for example if a running experiment is consuming too many resources or takes too long to complete.

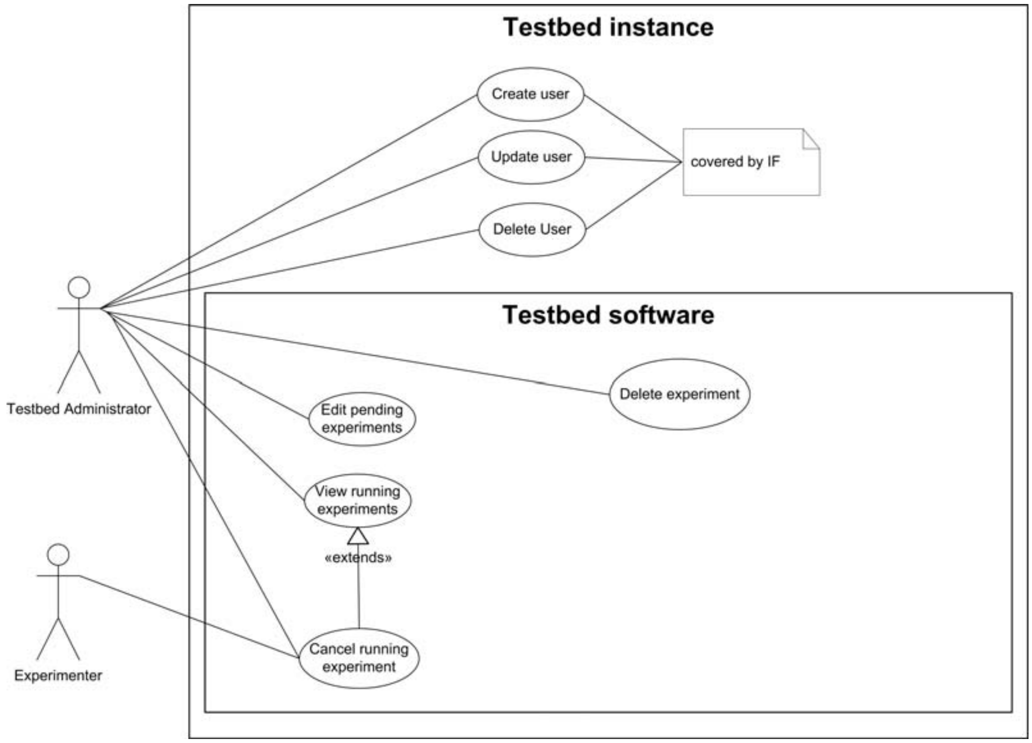


Figure 4.4: Testbed administration

Figure 4.4 depicts some of the administration use cases and shows the relationship between the user management use cases covered by the Interoperability Framework and those that form part of the testbed software. We then provide the use cases that are most relevant for administration:

1. View running experiments,
2. Cancel running experiment,
3. Delete an experiment, and
4. View available services.

View running experiments

Scope	Testbed Software
Level	User Goal
Actors	Testbed administrator
Preconditions	The administrator is logged in.
Trigger	The administrator selects the according function in the main menu.
Success guarantees	All running experiments are displayed.
Main Success Scenario	
1	System displays a window with the running experiments, showing at least • the owner of the experiment, • the start time, • percentage of CPU usage, and • memory usage.
2	The administrator can cancel any running experiment.
Exceptions	
-	none

Cancel running experiment

Scope	Testbed Software
Level	User Goal
Actors	Testbed administrator
Preconditions	The administrator is viewing the list of running experiments.

Trigger	The administrator activates this function while viewing running experiments, probably because an experiment is consuming too much resources or because of a user request.
Success guarantees	The selected experiment is cancelled and does not further consume any resources.
Main Success Scenario	
1	The system asks for confirmation and reasoning.
2	The administrator may enter a reason in text form and confirms the cancellation.
3	The system cancels the selected experiment and returns to the list of running experiments.
Exceptions	
-	none

Delete experiment

Scope	Testbed Software
Level	User Goal
Actors	Testbed administrator
Preconditions	The administrator has selected an experiment from the experiments registry.
Trigger	The administrator selects the according function in the menu, probably because he received a request to delete an experiment.
Success guarantees	The selected experiment is deleted.
Main Success Scenario	
1	The system presents a very short overview of the main properties of the experiment and asks for confirmation.
2	The administrator confirms the deletion of the selected experiment.
3	The system deletes the experiment details, settings, and results.
Exceptions	
-	none

4.4 Summary

In this chapter, we described the use case model for the digital preservation testbed. The main blocks of functionality we considered are:

- **Performing experiments.** – This is the core part of the testbed. Users need to be able to specify in detail the kind of experiment they want to conduct, the resources they need and the services they want to use. They need notification mechanisms for events such as experiment completion, and a thoroughly specified workflow to allow for traceability of experiment definition and reproducibility of results.

The process of performing a testbed experiment can be divided further into the following main chunks:

- *Experiment definition*,
- *Tool Configuration*,
- *Test data definition*,
- *Approving, scheduling, and running experiments*, and
- *Evaluating experiments.*

We described the use cases modelling this process in detail. We furthermore put the testbed workflow in context to **preservation planning** and introduced the main use case of the planning tool which will be deployed onto the testbed machine and accessible from the testbed software.

- **Access benchmark data.** – A central part of experiments is defining the data set on which the experiment should execute. As a data repository is therefore an important part of the testbed system, the software will provide an interface to it and enforce the corresponding security constraints.
- **Access experiment results.** – Users want to compare their results with previous experiment results and to search for previous experiments in order to analyse the respective data and decide if they need to conduct their own experiment. Furthermore, they need to be able to download experiments as complete packages for documentation purposes and want to reuse existing settings for future experiments. An experiment registry provides this functionality.
- **Management tasks** include the handling of running experiments and the deletion of experiment data.

The final chapter will summarise the work described in this thesis and provide an outlook to future work.

Chapter 5

Summary and Outlook

In this thesis, we proposed a functional model of a testbed for evaluating digital preservation approaches.

The first two chapters introduced the concepts of digital preservation in general and the challenges of testing and evaluating alternative strategies for preserving digital content.

Evaluating digital preservation strategies to arrive at well-informed, accountable, and in particular well-documented and traceable decisions on which path to pursue to ensure the survival and accessibility of digital objects, is a challenging task. Preservation planning tools are of great help in documenting the criteria of a specific institution that define the quality and fitness-for-purpose of particular solutions. However, they also rely on objective measurements and comparative benchmarking data on detailed characteristics of tools.

A digital preservation benchmark thus is urgently needed in the digital preservation community, and several related efforts are being undertaken to arrive at this goal. The DELOS project is working on criteria on an 'open testbed digital object corpus'. Section 2.3.1 provided an overview of these efforts. In relation to this, the PLANETS Testbed will form a stable platform to evaluate preservation services in a consistent, comparable, and well-documented way.

Chapter 2 further introduced the requirements specification methodology that we used to describe the functional model, namely use cases.

Chapter 3 laid out the foundations of this testbed. We described its predecessor, the Dutch Testbed, and its context, the PLANETS project with its distributed service architecture. We then gave an overview of the core functional tasks of the testbed.

Chapter 4 described in detail the functional usage model of the PLANETS Testbed. We characterised the user profiles and roles that will be interacting with the system and provided use case descriptions grouped by functionality and stages in the experiment process.

Specifically, we provided the usage model covering the following areas:

1. Experiment definition,
2. Configuration of tools to be evaluated,
3. Data set configuration,
4. The process of submitting and executing experiments,
5. The evaluation of experiments,
6. Access to the experiment registry,
7. Reuse of experiment setups, and
8. Administration.

The Testbed project is currently designing and implementing the system. The launch of the first version of the testbed software embedded in the Interoperability Framework will mark a significant step forward in direction of establishing benchmark procedures for digital preservation, which will hopefully bring the community of researchers and practitioners considerably further towards their goal of ensuring long-term access to our cultural and scientific heritage.

Bibliography

[And05] Bjarne Andersen. The dk-domain: In words and figures. Technical report, The State and University Library, Århus, Denmark, 2005.

[Arc] The National Archives. PRONOM - the technical registry. `http://www.nationalarchives.gov.uk/pronom/`, August 2006.

[Art06] Arts and Humanities Data Service. DAAT: Digital Asset Assessment Tool. Website, 2006. `http://ahds.ac.uk/about/projects/daat`, accessed: February 19, 2007.

[bam07] Archiving the Avant Garde: Documenting and Preserving Digital / Variable Media Art. Website, February 2007. `http://www.bampfa.berkeley.edu/about_bampfa/avantgarde.html`, accessed March 3, 2007.

[Bla02] Rex Black. *Managing the Testing Process*. John Wiley & Sons, New York, 2nd edition, 2002.

[BSN$^+$06] Christoph Becker, Stephan Strodl, Robert Neumayer, Paul Wheatley, Peter Bright, and Eleonora Nicchiarelli. PLANETS testbed use case specification. Internal Project Deliverable, December 2006.

[CBC$^+$99] H.B. Chen, O. Bimber, C. Chhatre, E. Poole, and S. J. Buckley. eSCA: a thin-client/server/web-enabled system for distributed supply chain simulation. In *Proceedings of the 31st conference on Winter simulation*, Phoenix, Arizona, United States, 1999.

[CLI02] CLIR and LoC. *Building a National Strategy for Digital Preservation: Issues in DigitalMedia Archiving*. Council on Library and Information Resources Washington, D.C. and

Libraryof Congress, 2002. http://www.clir.org/pubs/reports/pub106/pub106.pdf, accessed February 5, 2007.

[Coc00] Alistair Cockburn. *Writing Effective Use Cases*. The Agile Software Development Series. Addison-Wesley, 2000.

[Com02] Computer Knowledge (CKNOW). FILExt: The file extension source. Webpage, June 2002. http://filext.com/, accessed: March 5, 2007.

[ERP04] ERPANET. The archiving and preservation of born-digital art workshop. Briefing Paper for the ERPANET workshop on Preservation of Digital Art, 2004.

[Flo05] Florida Centre for Library Automation. Recommended data formats for preservation purposes in the FCLA digital archive, June 2005. http://www.fcla.edu/digitalArchive/pdfs/recFormats.pdf, accessed February 5, 2007.

[GC04] Maria Guercio and Cinzia Cappiello. File formats typology and registries for digital preservation. Technical report, DELOS - Network of Excellence on Digital Libraries, 2004.

[Ger91] Anne Geraci. *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries*. The Institute of Electrical and Electronics Engineers Inc., 1991.

[Har05] Harvard University Library. Jhove - jstor/harvard object validation environment. Website, 2005. http://hul.harvard.edu/jhove, accessed February 7, 2007.

[HC04] Jane Hunter and Sharmin Choudhury. A semi-automated digital preservation system based on semantic web services. In *Proceedings of the Joint Conference on Digital Libraries, JCDL 2004*, Tucson, Arizona, June 2004. ACM.

[HC05] Jane Hunter and Sharmin Choudhury. PANIC - An integrated approach to the preservation of composite digital objects using semantic web services. In *Proc. 5th Int. Web Archiving Workshop (IWAW05)*, Vienna, Austria, September 2005.

[HC06] Jane Hunter and Sharmin Choudhury. PANIC - an integrated approach to the preservation of complex digital objects using semantic web services. *International Journal on Digital Libraries: Special Issue on Complex Digital Objects*, 6(2):174–183, April 2006.

[HVD+04] Hans Hofman, Remco Verdegem, Michael Day, Andreas Rauber, Manfred Thaller, and Seamus Ross. DELOS deliverable 6.1: Framework for testbed for digital preservationexperiments. Technical report, DELOS Network of Excellence, November 2004.

[HVDDVEM05] J.R. Hoeven, R.J. Van Der Diessen, and K. Van En Meer. Development of a universal virtual computer (UVC) for long-term preservation of digital objects. *Journal of Information Science*, Vol. 31 (3):196–208, 2005.

[JB02] Maggie Jones and Neil Beagrie. *Preservation Management of Digital Materials: A Handbook.* Digital Preservation Coalition, 2002. `http://www.dpconline.org/graphics/handbook`, accessed: March 1, 2007.

[JCJv93] I. Jacobson, M. Christerson, P. Jonsson, and G. Övergaard. *Object-oriented software engineering: a use-case driven approach.* Addison-Wesley, 1993.

[JEJ94] I. Jacobson, M. Ericsson, and A. Jacobson. *The Object Advantage.* ACM Press Books, 1994.

[Koe05] Paul Koerbin. Report on the crawl and harvest of the whole australian web domain undertaken during june and july 2005. Public Report, October 2005. `http://pandora.nla.gov.au/documents/domain_harvest_report_public.pdf`, accessed: March 1, 2007.

[Lew] David Lewis. Reuters-21578 test collection. `http://www.daviddlewis.com/resources/testcollections/reuters21578`, accessed: March 1, 2007.

[Lib] Library of Congress. Sustainability of digital formats planning for library of congress collections. `http://www.digitalpreservation.gov/formats`, accessed: February 8, 2007.

[LKR+00] Gregory W. Lawrence, William R. Kehoe, Oya Y. Rieger, William H. Walters, and Anne R. Kenney. Risk management of digital information: A file format investigation. CLIR Report 93, CLIR, June 2000.

[Lor02] R.A. Lorie. The UVC: A method for preserving digital documents: Proof of concept. Technical report, IBM and Koninklijke Bibliotheek (KB) Netherlands, 2002.

[Mel03] Phil Mellor. Camileon: Emulation and BBC domesday. *RLG DigiNews*, 7(2), April 2003.

[Mil] Miodrag Milanovic. File format encyclopaedia. `http://pipin.tmd.ns.ac.yu/extra/fileformat/`, accessed: October 31, 2006.

[Nat] National Institute of Standards and Technology. Text retrieval conference (TREC). `http://trec.nist.gov/`, accessed January 8, 2007.

[NBL+07] Robert Neumayer, Christoph Becker, Thomas Lidy, Andreas Rauber, Eleonora Nicchiarelli, Manfred Thaller, Michael Day, Hans Hofman, and Seamus Ross. Development of an open testbed digital object corpus. DELOS Digital Preservation Cluster, Task 6.9, March 2007.

[NMB02] E. Nasr, L. McDermid, and G. Bernat. Eliciting and specifying requirements with use cases for embedded systems. In *Proceedings of the 7th International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS'02)*, pages 350–357, January 2002.

[OP04] Gunnar Övergaard and Karin Palmkvist. *Use Cases : Patterns and Blueprints*. The Software Patterns Series. Addison-Wesley Professional, November 2004.

[otN01] Nationaal Archief of the Netherlands. Digital preservation testbed: Research framework. Technical report, Nationaal Archief of the Netherlands, 2001.

[OvDvW04] Erik Oltmans, Raymond J. van Diessen, and Hilde van Wijngaarden. Preservation functionality in a digital archive. In *Proceedings of the Joint Conference on Digital Libraries,* *June 7-11, Tuscon, Arizona, USA*, pages 279–286. ACM, 2004.

[Pet03] Jo Pettitt. *PRONOM - Field Descriptions*. The National Archives, Digital Preservation Department, 2003. `http://www.records.pro.gov.uk/~pronom`.

[PLA06a] PLANETS. Annex I – description of work. FP6 Contract No. 033789, July 2006.

[PLA06b] PLANETS. Initial design specification for the interoperability framework. Internal Project Deliverable, December 2006.

[RAW02] A. Rauber, A. Aschenbrenner, and O. Witvoet. Austrian on-line archive processing: Analyzing archives of the world wide web. In *Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2002)*, pages 16–31, Rome, Italy, September 16-18 2002. Springer.

[RB99] Jeff Rothenberg and Tora Bikson. Carrying authentic, understandable and usable digital records through time. Technical report, Report to the Dutch National Archives and Ministry of the Interior, 1999.

[RB06] Thomas Reichherzer and Geoffrey Brown. Quantifying software requirements for supporting archived office documents using emulation. In *Proceedings of the 6th Joint Conference on Digital libraries (JCDL 2006)*, pages 86–94, New York, NY, USA, 2006. ACM Press.

[RLRS05] David S. H. Rosenthal, Thomas Lipkis, Thomas S. Robertson, and SethMorabito. Transparent format migration of preserved web content. *D-Lib Magazine*, 11(1), January 2005.

[Rot99] J. Rothenberg. *Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation*. Council on Library and Information Resources, January 1999. `http://www.clir.org/pubs/reports/rothenberg/contents.html`.

[RR04] Carl Rauch and Andreas Rauber. Preserving digital media: Towards a preservation solution evaluation metric. In *Proceedings of the 7th International Conference on Asian Digital Libraries (ICADL 2004)*, pages 203–212, Shanghai, P.R. China, December 13-17 2004. Springer.

[RRL+05] David S. H. Rosenthal, Thomas Robertson, Tom Lipkis, Vicky Reich, and Seth Morabito. Requirements for digital preservation systems. *D-Lib Magazine*, 11, November 2005. `http://www.dlib.org/dlib/november05/rosenthal/11rosenthal.html`.

[RSW02] Tony Rose, Mark Stevenson, and Miles Whitehead. The reuters corpus volume 1 - from yesterday's news to tomorrow's language resources, 2002.

[RSW05] Tony Rose, Mark Stevenson, and Miles Whitehead. The reuters corpus volume 2 - multilingual corpus, 2005.

[SBN+07] Stephan Strodl, Christoph Becker, Robert Neumayer, Andreas Rauber, Eleonora Nicchiarelli Bettelli, Max Kaiser, Hans Hofman, Heike Neuroth, Stefan Strathmann, Franca Debole, and Giuseppe Amato. Evaluating preservation strategies for electronic theses and dissertations. In *DELOS Conference on Digital Libraries*, pages 297–305, Pisa, Italy, February 13-14 2007.

[SBNR07] Stephan Strodl, Christoph Becker, Robert Neumayer, and Andreas Rauber. How to choose a digital preservation strategy: Evaluation a preservation planning procedure. In *Proceedings of the ACM IEEE Joint Conference on Digital Libraries (JCDL'07)*, June 2007.

[SJP02] F.T. Sheldon, K. Jerath, and O. Pilskalns. Case study: B2B e-commerce system specification and implementation employing use-case diagrams, digital signatures and XML. In *Proceedings of the 4th International Symposium on Multimedia Software Engineering (MSE'02)*, pages 106–113, December 2002.

[SM86] Gerald Salton and Michael J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, NY, 1986.

[Smi05] Mackenzie Smith. Eternal bits: How can we preserve digital files and save our collective memory? *IEEE Spectrum*, 42(7), July 2005.

[SRR$^+$06] Stephan Strodl, Andreas Rauber, Carl Rauch, Hans Hofman, Franca Debole, and Giuseppe Amato. The DELOS testbed for choosing a digital preservation strategy. In *Proceedings of the 9th International Conference on Asian Digital Libraries, ICADL 2006*, pages 323–332, Kyoto, Japan, November 27-30 2006. Springer.

[SV04] Jacqueline Slats and Remco Verdegem. Practical experiences of the dutch digital preservation testbed. *VINE: The Journal of Information and Knowledge Management Systems*, 34(2):56–65, 2004.

[TBS$^+$03] Robert Tansley, Mick Bass, David Stuve, Margret Branschofsky, Daniel Chudnov, Greg McClellan, and MacKenzie Smith. Managing resources and services: The dspace institutional digital repository system: current functionality. In *Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 87 – 97. ACM, 2003.

[The06] The PLANETS Project. Planets flyer. `www.planets-project.eu/docs/comms/Planets\_flyer\_2006.pdf`, accessed March 1, 2007, December 2006.

[UNE03] UNESCO. UNESCO charter on the preservation of digital heritage, October 17 2003. Adopted at the 32nd session of the General Conference of UNESCO.

[var07] Variable media network. Website, February 2007. `http://www.variablemedia.net/`, accessed March 3, 2007.

[vdHvW05] Jeffrey van der Hoeven and Hilde van Wijngaarden. Modular emulation as a long-term preservation strategy for digital objects. In *5th International Web Archiving Workshop (IWAW05)*, 2005.

[Web05] Colin Webb. *Guidelines for the Preservation of the Digital Heritage*. Information Society Division United Nations Educational, Scientific and Cultural Organization (UNESCO) – National Library of Australia, 2005. `http://www.unesco.nl/images/guidelines.pdf`.

[WO04] H. Van Wijngaarden and E. Oltmans. Digital preservation and permanent access: the UVC for images. In *Proceedings of Imaging Science & Technology Archiving Conference*, San Antonio, USA, April 13 2004. `http://www.kb.nl/hrd/dd/dd_links_en_publicaties/publicaties/uvc-ist.pdf`, accessed March 20, 2007.