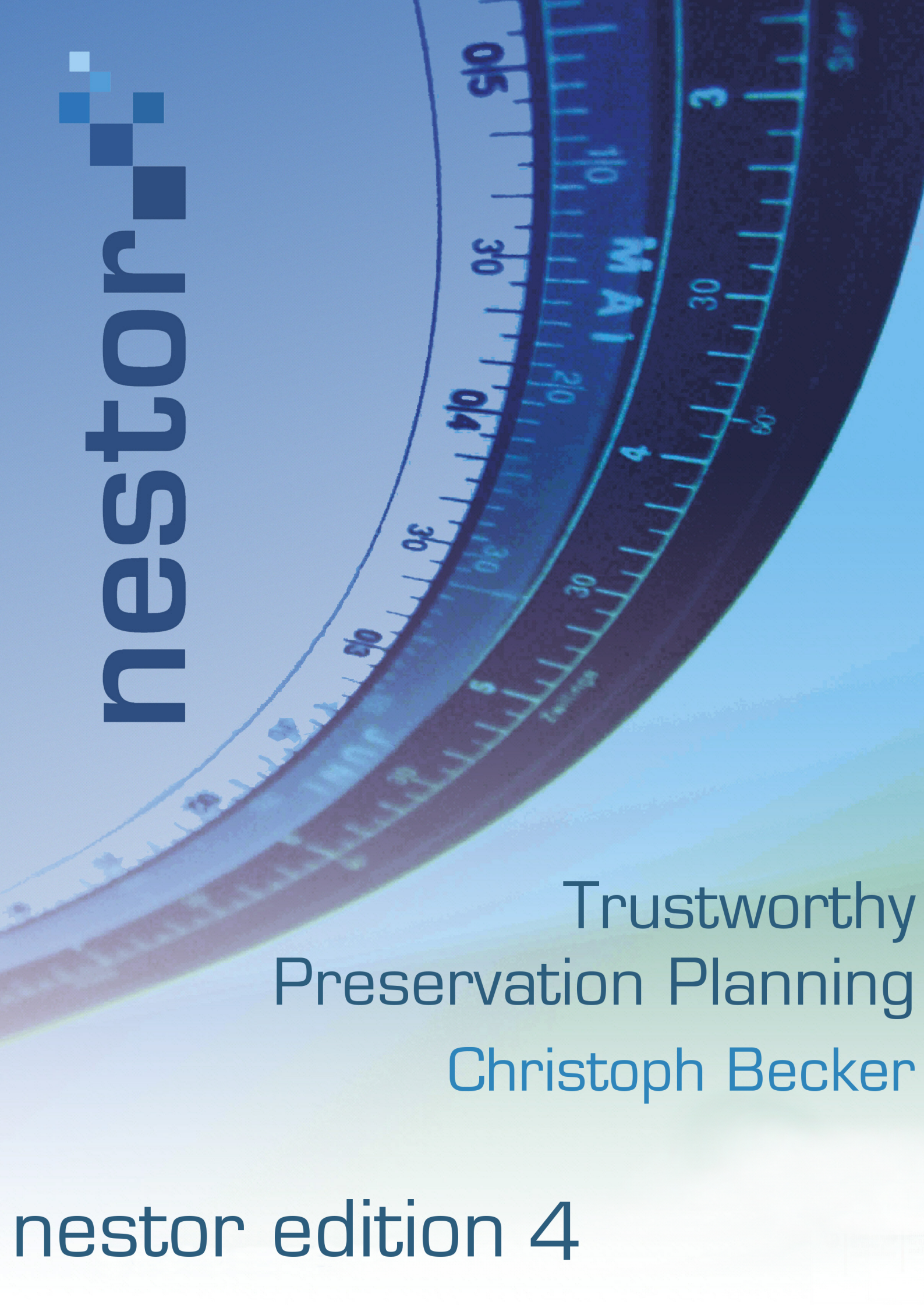


Trustworthy Preservation Planning

Christoph Becker

nestor edition 4

Trustworthy Preservation Planning

Christoph Becker

nestor edition 4

Herausgegeben von

nestor - Kompetenznetzwerk Langzeitarchivierung und
Langzeitverfügbarkeit Digitaler Ressourcen für Deutschland

nestor - Network of Expertise in Long-Term Storage
of Digital Resources

http://www.langzeitarchivierung.de

nestor Kooperationspartner:

- Bayerische Staatsbibliothek
- Deutsche Nationalbibliothek
- FernUniversität Hagen
- Georg-August-Universität Göttingen / Niedersächsische Staats- und Universitätsbibliothek Göttingen
- Humboldt-Universität zu Berlin
- Landesarchiv Baden-Württemberg
- Stiftung Preußischer Kulturbesitz / SMB - Institut für Museumsforschung
- Bibliotheksservice-Zentrum Baden-Württemberg
- Institut für Deutsche Sprache
- Computerspiele Museum Berlin
- Goportis
- PDF/A Competence Center

URN: urn:nbn:de:0008-2011061603

[http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:0008-2011061603]

Die Schriftenreihe nestor edition präsentiert ausgewählte wissenschaftliche Arbeiten mit dem Schwerpunkt Langzeitarchivierung.
Die Reihe wird in loser Folge von nestor – Kompetenznetzwerk Langzeitarchivierung herausgegeben. Damit entsteht ein Forum, in dem Beiträge zu verschiedenen Aspekten der digitalen Langzeitarchivierung einer breiten Öffentlichkeit zugänglich gemacht werden.
Die Arbeiten werden von ausgewiesenen Experten aus den jeweiligen Fachgebieten für die nestor edition gezielt ausgewählt, wenn sie einen besonderen Beitrag zu wichtigen Themenfeldern oder zu neuen wissenschaftlichen Forschungen auf dem Gebiet leisten.

Bemerkungen zu dieser Publikation, aber auch Vorschläge für die Aufnahme weiterer Beiträge in der Edition gerne an: VL-nestor@d-nb.de

Für die Partner des Projekts nestor – Kompetenznetzwerk Langzeitarchivierung
Reinhard Altenhöner und Natascha Schumann
Deutsche Nationalbibliothek

DISSERTATION

Trustworthy Preservation Planning

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften
unter der Leitung von

ao.univ.Prof. Dr. Andreas Rauber
E188
Institut für Softwaretechnik und Interaktive Systeme

eingereicht an der Technischen Universität Wien
Fakultät für Informatik
von

Dipl.-Ing. Mag.rer.soc.oec. Christoph Becker

Wien, am *12. Mai 2010*

Kurzfassung

Die Aufgabe der digitalen Langzeitarchivierung ist es, die Risiken abzuwehren, die die Vielzahl existierender digitaler Materialien auf den Ebenen der Datenströme, der Logik und der Semantik bedrohen und die langfristige Verfügbarkeit und Verständlichkeit dieser Materialien in Frage stellen. Das erklärte Ziel besteht darin, langfristige, sichere und authentische Speicherung sowie den vertrauenswürdigen Zugriff auf digitale Inhalte in einer verwendbaren Form für eine definierte Benutzergruppe sicherzustellen. Das erfordert auf Grund der konstanten Veränderungen der verwendeten Technologien kontinuierliche Aktionen zur Bewahrung der Objekte und zur Sicherstellung ihrer Lesbarkeit nach dem Ende der Verfügbarkeit der ursprünglichen technischen Umgebung, die zur Herstellung und Wiedergabe benutzt wurde. Solche Aktionen dienen daher entweder der Wiederherstellung einer äquivalenten Umgebung (Emulation) oder der (wiederholten) Konvertierung des Objektes in eine Repräsentationsform, die mit aktuellen Umgebungen kompatibel ist.

Grundsätzlich steht meist eine Vielzahl potentieller Aktionen zur Verfügung. Deren Qualität variiert jedoch je nach eingesetzter Software stark, die Eigenschaften digitaler Objekte unterscheiden sich je nach dem Typ der Inhalte, und die Arten der Verwendung und die entsprechenden Anforderungen variieren je nach Zielgruppe und Zugriffsszenarien. Risikotoleranz, Präferenzen, Kosten und Einschränkungen technischer und organisationsbedingter Art schwanken je nach der betrachteten Sammlung von Inhalten, der verantwortlichen Organisation und ihrer Umgebung. Weiters sind all diese Faktoren konstanten Verschiebungen ausgesetzt, die es zu erkennen und zu behandeln gilt.

Die Mission der Planung von vertrauenswürdiger Langzeitarchivierung besteht also darin, authentischen Zugriff für die Zukunft sicher zu stellen, indem die richtigen Aktionen definiert werden, um bestimmte Inhalte zu bewahren. Das Kernproblem dieser Planung ist eine domänenspezifische Variante eines bekannten Problemes der Softwareherstellung – der Selektion einer optimalen Komponente zur Erfüllung spezifischer Funktionen und ihre Integration in ein Software-System. Folgende Forschungsfragen ergeben sich dabei: (1) Wie kann man die für eine bestimmte Situation optimale Aktion zur Langzeitarchivierung auswählen? (2) Wie kann man dabei vertrauenswürdige Planung sicherstellen? (3) Wie kann man erreichen, dass die Entscheidungsprozesse heutigen und künftigen Anforderungen entsprechend skalieren?

Diese Dissertation beschreibt einen systematischen Ansatz zur Planung von Langzeitarchivierung. Dieser Ansatz besteht aus einer Entscheidungsmethode für Situationen mit einer Vielzahl potentiell widersprüchlicher Kriterien. Diese Methode wird begleitet von einem konkreten Arbeitsprozess und einem Softwarewerkzeug, das die Erstellung von Archivierungsplänen für definierte Mengen von digitalen Objekten unterstützt. Richtlinien als

abstrakte Einflussfaktoren modellieren dabei bekannte Einschränkungen und dokumentieren die Präferenzen der entscheidungstreffenden Organisation. Planungsverantwortliche Entscheidungsträger evaluieren auf dieser Basis potentielle Aktionen und Komponenten auf empirische Weise, indem sie automatische Messungen in einer kontrollierten Umgebung durchführen und auf Grund der gesammelten Messdaten die Komponente auswählen, die die Anforderungen einer bestimmten Situation am Besten erfüllt.

Diese Arbeit stellt zu diesem Zweck eine verteilte Software-Architektur zur Planungsunterstützung für Langzeitarchivierung vor, in der Planung, Aktionen, und Charakterisierung eng gekoppelt und integriert sind. Das Herzstück dieser Architektur bildet das Planungswerkzeug *Plato* (Planning Tool). Diese Software implementiert die Planungsmethode und erstellt solide, automatisch dokumentierte Archivierungspläne. Das Werkzeug hat in der weltweiten Gemeinschaft der Langzeitarchivierung signifikantes Interesse erfahren und wurde bereits zur produktiven Entscheidungsfindung in mehreren nationalen Institutionen eingesetzt.

Die Arbeit diskutiert Beispiele, in denen der Ansatz auf tatsächliche Probleme angewandt wurde, erforscht Einschränkungen und Kernprobleme des Ansatzes und identifiziert insbesondere die Schlüsselfrage der Evaluierung. Eine Analyse von Einflussfaktoren, die berücksichtigt und evaluiert werden müssen, führt zu einer Kategorisierung von Entscheidungskriterien in einer Taxonomie. Es wird gezeigt, dass ein Großteil der Kriterien durch automatische Messungen in einer kontrollierten Umgebung bei realistischen Bedingungen evaluiert werden kann. Es wird weiters demonstriert, dass kontrollierte Experimente und automatische Messungen die Wiederholbarkeit von Entscheidungen substantiell verbessern. Dadurch wird der Aufwand der Evaluierung von Komponenten reduziert und die Skalierbarkeit deutlich verbessert. Die automatische Messung unterstützt außerdem die Vertrauenswürdigkeit von Entscheidungen, da ausführliches Beweismaterial in einer wiederholbaren und nachvollziehbaren Weise produziert wird und dieses in standardisierter und vergleichbarer Form dokumentiert ist.

Abstract

The mission of digital preservation is to overcome the obsolescence threats that digital material is facing on the bitstream, the logical, and the semantic level, and to provide continued, authentic long-term storage and access to digital objects in a usable form for a specific user community. This requires preservation actions to be carried out when the original environment of digital objects is unavailable, to either recreate it (emulation) or transform the objects' representation into a form usable in a new enviroment (migration). A variety of preservation actions exist. Quality varies across tools; properties vary across content; usage and requirements vary across users and scenarios; risk tolerances, preferences, costs, and constraints vary across collections, organisations, and environments. Finally, all of these factors are subject to constant shifts that have to be detected and handled.

The mission of preservation planning is to ensure authentic future access for a specific set of objects by defining the actions needed to preserve it. The core problem of preservation planning is a domain-specific instance of component selection and can be correspondingly reformulated and modelled. The arising research questions are threefold: (1) How can we select the optimal preservation action for a given setting? (2) How can we ensure trustworthy preservation planning? (3) How can we ensure that decision processes scale up?

This thesis describes a systematic framework for preservation planning, comprising a multi-objective decision making method, workflow and tool for creating preservation plans for sets of digital objects. Policies as high-level influence factors model environmental constraints and specify organisational preferences. Preservation planners empirically evaluate potential action components by applying automated measurements in a controlled environment and select the component that is optimal with respect to the particular requirements of a given setting. We present a distributed architecture for preservation planning which integrates planning, actions, and characterisation, with the planning tool Plato at its core. The tool implements the planning method and creates solid, well-documented preservation plans. It has experienced significant uptake in the digital preservation community.

We describe examples applying the approach to a number of real-world business decisions, discuss limitations of the approach, and identify a key challenge of evaluation. We further analyse the influence factors to be captured and evaluated and categorise them in a taxonomy. We show that a majority of the criteria can be evaluated by applying automated measurements under realistic conditions, and demonstrate that controlled experimentation and automated measurements can be used to substantially improve repeatability of decisions. This reduces the effort needed to evaluate components, enables scalability, and supports trust in the decisions because extensive evidence is produced in a repeatable and reproducible way and documented along with the decision in a standardised and comparable form.

Acknowledgements

This thesis would not have been possible without the support of my colleagues, friends, and family.

Thanks go to Andi Rauber for countless things, starting with the invitation to join an exciting team; for a lot of critical and fruitful discussions, including heated arguments on crucial issues; for encouraging motivation and offers to take on new challenges; and for almost 24/7 support in the midst of an unbelievably busy schedule.

A lot of the work described here would have never been possible without the brilliant collaboration I enjoyed (and hope to continue enjoying!) – above all with Hannes and Michael on the planning tool and all the ideas and projects related to it, but also with Mark, Stephan and the *ateam*.

I would never have had the strength to do all this without my friends, to whom I am deeply indebted... and, last but most important, my parents, whom I want to thank for everything.

Part of this work was supported by the European Union in the 6th Framework Program, IST, through the PLANETS project, contract 033789. The discussions with partners within and outside of Planets, and the critical feedback we received, were a constant motivation to improve on our methods and tools. Special thanks go furthermore to José Borbinha for taking the time to review my work in detail on such a short schedule.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The fragile nature of digital information

The last decades have made digital objects the primary medium to create and exchange information. Digital objects increasingly contain essential parts of our cultural, intellectual and scientific heritage; they form a central part of our economy, and their ubiquity is increasingly shaping our private lives. Digital photography has long exceeded the analog pendant in popularity, and more and more artists focus on digital media in their work. Ten years ago there was a distinction between *Mail* and *Email*; today it is more common to differentiate between *Mail* and *Snail mail*, since the electronic version of the traditional messaging system is now generally seen as the more obvious form of communication.

The amount of information created each year is soaring. In 2000, the world produced between one and two exabytes of new information [VL00], i.e. two to three million Terabytes; in 2007, technology research firm IDC estimated that in 2006 the amount of data produced had grown to 161 Exabytes and would further explode to 988 Exabytes by 2010 [Gau07]. In fact, by the end of 2009 the digital universe has already grown to 800 Exabytes and will likely exceed 1.2 zettabytes, i.e. 1.2 million petabytes, by the end of 2010 [GR10].

This shift in the way humankind creates and exchanges information has led to a dominance of digital objects in today's information landscape. It has allowed us to produce, store, search, consume and connect unprecedented amounts of data in ways never imagined before. However, the digital age has also introduced an entirely new problem: the lack of longevity of digital objects. Due to the fast changes in technologies, digital documents have a short lifespan before they become obsolete. The ever-growing complexity and heterogeneity of digital file formats together with rapid changes in underlying technologies have posed extreme challenges to the longevity of information. So far, digital objects are inherently ephemeral. Memory institutions such

as national libraries and archives as well as space data archives [Bla90] were amongst the first to approach the problem of ensuring long-term access to digital objects when the original software or hardware to interpret them correctly becomes unavailable [UNE03]. Today, awareness of the issue is raising, as the topic is presenting a challenge to key players in a wide range of domains such as manufacturing, finance, pharmaceutical companies, medicine, and e-Science [Man10].

The longevity of digital objects used to be something taken for granted by many, until in the last decade several instances of spectacular data loss drew the public's attention to the fact that digital objects do not last forever. One of the best known case studies in digital preservation, the rescue of BBC Domesday [Mel03], is a prominent example of almost irrecoverable data loss due to obsolescence of hardware and software capable of reading and interpreting the content. Large amounts of money and effort were required to make the data accessible again and adequately preserve them for the future. Recently, a survey among professional archivists underlined the growing awareness of the urgency of digital preservation [The07]. This awareness has led to the development of various approaches that deal with the question of preserving digital objects over long periods of time.

The primary reason why digital objects become unaccessible lies within their very nature. While analog objects such as photographs or books directly represent the content, digital objects are often useless without the technical environment they have been designed for. In contrast to traditional non-electronic objects such as books or photographs which immediately *are* the content, CAD drawings cannot be opened and read, a simulation cannot be re-run and re-evaluated, sensor data cannot be interpreted without a suitable hardware, software and documentation environment. A digital object always needs an environment to render, or *perform*, it. These environments keep evolving and changing at a rapid pace, which brings about the problem of digital continuity. The constant changes in IT environments render objects incompatible within years and thus challenge the longevity of digital information. A variety of reasons cause obsolescence, e.g. media failure, file formats and tools becoming obsolete, or the loss of necessary metadata. Especially for born-digital material this often means that the contained information is lost completely.

Digital content is threatened on three levels:

1. On a physical level, storage media are much more volatile than many of their analog counterparts. While properly stored papyrus lasts thousands of years, a modern hard disk will almost inevitably fail within a decade. Even if the fragile mechanics do not break, the interfaces of computer systems change with a pace that renders technologies incompatible within years and makes replacements necessary. Still, storage system technology has advanced significantly and is able to provide

highly reliable redundant storage systems and bitstream preservation strategies based on media migration and refreshment. The problem is mostly one of optimising efficiency and minimising risks and costs. However, digital longevity on another level is a problem area that continues to pose largely unsolved challenges:

2. On a logical level, the actual meaning of bitstreams is encoded in file formats to be interpreted by certain software, installed in certain operating systems. These in turn are dependent on specific hardware. The thus-created dependency networks are very hard to preserve or reconstruct ex-post. This is the main focus area of this thesis.

3. On a semantic level, the understandability of content depends on semantic connotations of terms and concepts that are subject to change over time. Access patterns and modes of interaction with computers undergo dramatic changes within short periods of time. For instance, very few computer users nowadays are fluent in handling the *telnet* email clients that were the standard mode of reading emails just a few years ago. This semantic level, however, applies not only to digital content: For example, handling an old Historical Schellack 78rpm record or even a magnetic tape recorded at the end of the last century will prove a difficult task for many people today; and the semantics of terms may shift over time, as has been the subject of linguistic studies for many decades [Blo33, Ull57, Ull62]. We will thus not cover these aspects in our studies.

While traditional non-electronic objects have to be saved from gradually fading away, the life curve of digital objects usually is cut off sharply: Incompatible environments or the inability to recognise the format an object is kept in will often mean that the object is lost entirely. For content which was not digitised from paper, but born digitally, this loss is irrecoverable.

Digital preservation thus denotes the efforts to preserve digital content for a given purpose over long periods of time, after the original technical environment has become obsolete. The two dominant types of *preservation actions* taken to keep digital content alive today can be divided along this line. While migration transforms the objects to more widely accessible representations, emulation creates a technical environment where the objects can be rendered. Consider a collection of electronic documents created years ago on an old operating system running a now obsolete version of Microsoft Word. One could convert these documents to standardised formats such as Open Document Format ODF [ISO06] or the archival PDF standard PDF/A [ISO04a]. The latter would sacrifice the possibility to comfortably edit the documents for a file format that is widely supported and considered stable. Current software environments can then be used to access, view, and

print the document, providing users a familiar environment that they are able to use with ease.

On the other hand, one could emulate the obsolete technical environment and use the original software to access the objects in an authentic way. This could retain the look-and-feel of the original objects and provide access in much the same way as the authors were originally interacting with the content. However, even emulation proves challenging. For example, it might not be possible to print the documents on modern printers, and users might miss functionality such as copy-and-paste that they have come to expect from document editors.

If everyone producing or consuming information in digital form would agree on a standardised format for every type of content, and these formats would never change, the problem of digital preservation would hardly exist. While this is certainly not realistic nor desired, standardisation is nevertheless an important consideration, and an important part of ongoing efforts in many large international projects is the outreach to vendors for advocating document engineering technologies for sustainable documents. The effects can be seen in standards such as PDF/A [ISO04a], the Open Document Format (ODF) [ISO06], or MPEG-7 [ISO02]. However, many objects exist and many more are created every day that face the threats of obsolescence. Hence, ex-post actions for preserving access to content are necessary.

Preserving authentic records also means being able to prove authenticity [GSE00, RB99], but creating new manifestations of digital files in different representation formats always incurs the risk that parts of the content are not converted correctly. Hence, when migrating digital files, keeping the original bitstreams as a fallback strategy is common practice. However, having access to the original bitstreams does not guarantee that they are still legible in the future.

1.2 Problems and Research Questions

Evaluating preservation actions. –Various migration tools are available for converting objects in standard file formats such as office documents to representations that are considered more stable. The picture is less positive for more exotic and complex compound objects. However, even within migration tools for office documents, variation regarding the quality of conversion is very high. Some tools fail to preserve the proper layout of tables contained in a document; others miss footnotes or hyperlinks. Finding out which information has been lost during a conversion, and if this loss threatens the value of the object for a given purpose, is a very time-consuming task. Some losses might be acceptable, while others threaten the authenticity of documents. For example, if migrating the collection of Word documents mentioned above results in a loss of page breaks, this might be irrelevant if

the textual content is the only thing of interest. However, if there are page references in the text, this loss might be unacceptable.

While migration operates on the objects and transforms them to more stable or more widely adopted representations, emulation operates on the environment of an object, trying to simulate the original environment that the object needs, e.g. a certain processor or a certain operating system. This has the advantage of not changing the original objects, and of providing authentic access in much the same way than before. However, emulation is technically complex to achieve and hard to scale up to large amounts of data. Furthermore, users may have difficulties in using old software environments, and some functionality of newer systems, such as the copy-and-paste which is ubiquitous today, might not be available when relying on the original environment of an object. Moreover, as with migration, specific characteristics of an object may be lost due to incomplete or faulty emulation, or due to the impossibility to emulate certain aspects.

Multi-objective decision making. – The number of file viewers and file conversion tools for standard types of objects such as images or electronic documents is steadily increasing. Choosing the right treatment for a given set of objects is a crucial decision that needs to be taken based on a profound and well-documented analysis of the requirements and the performance of the tools considered. When deciding which solution is best suited for a given collection of objects and a specific purpose, the complex situations and requirements that need to be considered render this decision a complex task.

On the one hand, performance and quality of applicable tools vary; the requirements depend on a variety of factors including the content, the user communities, and the access scenarios; and each organisational setting poses distinct constraints. On the other hand, the decision maker has to achieve multiple competing objectives such as *minimise costs*, *ensure authenticity*, and *provide comfortable access*. When making these objectives operational, one must not distort the balance of the whole.

Component selection for preservation planning. – The task of selecting the optimal choice of action is one of the key responsibilities of the *preservation planning* function, which is at the heart of the Open Archival Information Systems model (OAIS) [ISO03]. The key result of such a preservation planning activity is a *preservation plan*. The structure of such a plan has not been clearly defined yet and needs to be specified. On the other hand, the selection problem can be seen as a domain-specific instance of the general problem of component evaluation and selection which has a long history in the areas of Software Engineering and Information Systems Design.

We seek a framework and system to enable decision makers select the preservation action component (or combination of components) which of all the available alternatives achieves the 'optimal' score with respect to multiple, potentially conflicting and initially ill-defined preservation goals.

This framework shall concretise vague requirements and rely on objective and repeatable measurements to ensure reproducibility. It shall further define action plans for deploying and using the selected component, and provide guidance on how to monitor the operational plan to ensure that any deviation from expected outcomes is noticed and can be reacted upon properly.

Trustworthy preservation. – Previous work has developed a workflow for evaluating potential preservation actions based on a customised variation of utility analysis [RR04]. While the workflow is useful and well applicable to the scenario, the selection is based on manual evaluation of the requirements, does not define a plan, and does not cover continuous monitoring. Until now, this selection is mostly an ad-hoc procedure with little tool support. This also implies that decisions that have been and are made are not transparent, hardly reproducible and often poorly documented. However, in complex environments with changing requirements, subjective human judgement of software quality and the reliance on declared capabilities of components cannot be considered sufficient evidence for trustworthy decision making, and cannot replace objective evidence as the basis of decision making. Accountability is widely seen as a major requirement for a trustworthy repository; and trustworthiness is probably the most fundamental requirement that a digital repository preserving content over the long term has to meet. For all decisions taken, we need full evidence of reasons and documentation to ensure auditable procedures that support trustworthiness.

As Terzis recently stated,

> ...the modern view of trust is that trustworthiness is a measureable property that different entities have in various degrees. Trust management is about managing the risks of interactions between entities. Trust is determined on the basis of evidence ... and is situational – that is, an entity's trustworthiness differs depending on the context of the interaction [Ter09].

If an entity's trustworthiness has to be validated in the context of an interaction, we need to do so in a controlled environment where the varying parameters are known and the outcomes repeatable, reproducible, and measurable.

Decision factors and automated measurement. – In order to improve decision making and trustworthiness, we need deep insight into the actual decision factors that need to be considered and the entities that they affect. We need to understand what exactly leads to rejection or acceptance of certain effects induced by preservation actions, what makes an action preferable to another, and what aspects of the environment and the organisational context have to be taken into account in decision making processes. Based on this, we can analyse the potential for automated measurements of these factors to improve repeatability and automation.

Automation and tool support for preservation planning. – To conduct repeatable preservation planning by evaluating potential components against multiple specific objectives, we need an architecture and tool support that structures the decision making procedures, collects measurements, documents decisions and provides the decision maker with far-reaching automation. The architecture and tool should guide the decision maker through the decision procedure and provide proactive support in the key areas of goal specification, definition of potential alternative actions, evaluation, and plan specification.

A controlled environment for experimentation. – The systematic evaluation of action components needs a controlled environment for the empirical validation of a component's behaviour in a certain scenario. This environment shall enable repeatable experiments and automated measurements to create a substantial evidence base and thus provide solid decision making support.

Based on these challenges, a number of research questions have been derived that are addressed in this theses. Particularly, these are:

RQ1: **How can we select the optimal preservation action for a given setting?**

This requires several challenges to be addressed, specifically

a. What are the constraints on the decision space?
b. What are the factors influencing the decision makers' preferences?
c. How can we model multiple competing objectives and requirements?
d. How should we evaluate software components?

RQ2: **How can we ensure trustworthy preservation planning?**
This entails in particular the following questions:

a. What are the requirements on trust that need to be addressed?
b. What decision steps and evidence need to be documented?
c. What are the aspects that a plan needs to address, and what are the elements needed to cover them?
d. How can we ensure reliable evaluation procedures and repeatable evidence?

RQ3: **How can we ensure that decision processes scale up?**

Given the increasing data volumes expected for the near future, decision making needs to be ready to scale up in terms of volume and complexity. This requires substantial automation in terms of decision making, monitoring, and measurement.

a. How can we automate decision making?

b. How can we integrate continuous monitoring?

c. Which properties can be measured automatically, and how?

d. How can we create a controlled environment for observing the behaviour of components in a reproducible way?

This thesis describes a solid and well-documented method, workflow and tool for creating trustworthy preservation plans for sets of digital objects. The method follows a variation of the utility analysis approach for supporting multi-criteria decision making procedures in digital preservation planning [RR04]. Preservation planners empirically evaluate potential action components, applying automated measurements in a controlled environment and selecting the most suitable one with respect to the particular requirements of a given setting. We present extensive tool support and discuss the substantial uptake the method and tool have experienced in the digital preservation community.

We further discuss a series of case studies. Based on these, we present an analysis of influence factors to be captured and evaluated, and show that a majority of the criteria can be evaluated by applying automated measurements in a controlled environment under realistic conditions. This not only reduces the effort needed to evaluate components, but also supports trust in the decisions because extensive evidence is produced in a repeatable and reproducible way and documented along with the decision in a standardised and comparable form.

1.3 Contributions

This work has naturally involved a network of collaborators and project partners. My specific contributions to the final picture are listed in the following.

1. I conducted a thorough review and analysis of the original method of utility analysis applied to the evaluation of preservation strategies, and have led the definition and concrete specification of a preservation plan [BKG+09].

2. I have conducted an extensive analysis of component selection literature and compared existing approaches. Based on this analysis, I could show how to reformulate the evaluation problem and generalise the evaluation and selection approach as a component selection scenario [BR09, BR10].

3. I developed an extension of the revised component evaluation and selection methodology to include the creation of preservation plans and discussed continuous monitoring and re-evaluation of plans [BKG+09].

4. I designed an architecture for integrating planning, actions, and measurements. Distributed information sources and preservation actions [BFK+08, BKK+09b] are integrated with planning, in-depth characterisation [BRH+08b, BRH+08a, Bec08] as well as quality-aware migration and performance measurements [BKK+09a, BKK+09b].

5. I led the development of a planning platform and tool [BKRH08] to support and automate the complete workflow and plan specification. The award-winning planning tool *Plato* is publicly available online[1], has a growing world-wide user base of over 560 accounts (as of May 2010) and is increasingly being used by large institutions for productive decision making [BKG+09, KRB+09].

6. I participated in or led case studies on preserving

   (a) electronic art [BKKR07],

   (b) electronic documents [BSN+07],

   (c) databases,

   (d) web pages [SBNR07], and

   (e) images [BKG+09].

7. Based on these case studies, I have defined a taxonomy of decision criteria relevant for the evaluation of preservation actions. Based on a quantitative evaluation of this taxonomy compared to real-world case studies, I discuss the coverage of automated measurements and show that a majority of the decision factors can be evaluated by applying automated measurements in a controlled environment [BR10].

1.4 Organisation

This section outlines the organisation of the thesis and provides references to the publications that comprise its main contributions.

- Chapter 2 introduces related work in the primary area of digital preservation as well as the key issue of component evaluation and selection. It introduces leading digital preservation initiatives and then focuses on the issue of trust in digital repositories. A discussion of preservation actions leads to an early case study on evaluating potential actions for preserving electronic documents, which was published in [BSN+07]. We review and analyse the original method of utility analysis applied to the evaluation of preservation strategies. Different approaches to

[1] `http://www.ifs.tuwien.ac.at/dp/plato`

systematic characterisation of objects as related to the central question of authenticity are discussed. These were originally published in [BRH+08a, BRH+08b].

- Chapter 3 presents an approach to defining well-structured preservation plans in a systematic way. We outline the requirements on preservation planning as derived from criteria on trustworthy digital repositories. We then show that the evaluation and selection problem can be reformulated as a component selection scenario [BR09]. Based on this, we introduce a framework for component evaluation and selection based on controlled experimentation [BR10]. We then describe in detail how the general framework translates into the 14-step preservation planning workflow that leads to the creation of a well-defined preservation plan for a specific set of digital objects [BKG+09].

- Chapter 4 introduces the planning tool Plato [BKRH08, BKR10] that we implemented as the core component of a distributed environment developed to support component selection and plan definition. This tool is integrated with the Planets suite of distributed preservation services and is increasingly used for productive decision making by a growing world-wide community. We give an overview of the variety of components for preservation actions that are integrated [BFK+08, BKK+09b].

- Chapter 5 describes a series of case studies on preservation planning [BKKR07, GBR08, BKG+09] and presents lessons learned. Based on this, we conduct a critical assessment of the planning approach, discussing criteria specification and requirements evaluation as a key issue.

- Chapter 6 presents an evaluation framework for collecting automated measurements in a controlled environment. We introduce a taxonomy of criteria and discuss examples of how each of the categories of the taxonomy can be measured. Examples of automated measurements are described. Distributed information sources and preservation actions [BFK+08, BKK+09b] are integrated with planning, in-depth characterisation [BRH+08b, BRH+08a, Bec08] as well as quality-aware migration and performance measurements [BKK+09a, BKK+09b].

  Based on a quantitative analysis of criteria in thirteen real-world case studies, it is shown that a majority of the criteria can be evaluated by applying automated measurements in a controlled environment under realistic conditions [BR10].

- Chapter 7 discusses limitations of the work and its wider applicability. We further give an outlook to implications on future work and open questions to be addressed in follow-up research.

Figure 1.1: Wordle of the text of this thesis

Chapter 2

Related work

2.1 Digital preservation

The mission of digital preservation is to overcome the obsolescence threats that digital material is facing on the bitstream, the logical, and the semantic level, and to provide continued, authentic long-term storage and access to digital objects in a usable form for a specific user community.

It is confined from digitisation, which is a challenging field in itself. Digitisation can be seen as a form of preservation for analog material, extending the life span of the intellectual content into the digital realm when the analog material decays. But in fact it creates the problem of digital preservation in that the digitised material has to be preserved in itself, as illustrated vividly by the Domesday case [Mel03].

The Digital Preservation Coalition[1] defines digital preservation as follows:

> ... the series of managed activities necessary to ensure continued access to digital materials ... it refers to all of the actions required to maintain access to digital materials beyond the limits of media failure or technological change [JB08]

In contrast,

> ...digitization is not preservation - it is simply a means of copying original materials. In creating a digital copy, the institution creates a new resource that will itself require preservation. Unlike microfilm and other preservation media whose longevity is assured relatively easily by proper storage, digital resources face many questions about how their continued existence, accuracy, and authenticity can be assured.[2]

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

[2] http://www.nedcc.org/resources/leaflets/6Reformatting/06PreservationAndSelection.php

The term 'digital preservation' often appears jointly with notions of 'long-term', referring to any period from a few to a thousand years. Generally speaking, *long-term* is any period long enough to pose challenges on the accessability of digital material due to changes in technologies and users that have an impact on the readability and understandability of material.

A number of research initiatives have emerged in the last decade as memory institutions, space agencies and data archives realised the challenge of the digital preservation problem [UNE03, SBR09]. Recently, a survey among archivists reemphasised the urgency of the matter [The07]. Thorough discussions are presented in [Web05] and [Thi02].

This chapter outlines related work in the areas of digital preservation and software systems. We first describe the most widely used model for a long-term archive, the Reference Model for an Archival Information System (OAIS), in Section 2.2. Section 2.3 discusses the question of trust in digital archives.

Preserving digital object requires *preservation actions* to be taken. In Section 2.4 we shortly describe the two dominating types of actions, migration and emulation, and previous work dealing with the evaluation of preservation actions, which is a predecessor of the approach and system presented here. We further discuss an early case study on evaluating preservation actions.

The central problem when defining preservation plans is one of selecting the optimal preservation action component. Section 2.5 gives an overview of related work in the area of Component Based System Development and Web Service selection. Systematic evaluation of action components for digital objects needs a systematic analysis of the objects themselves. Section 2.6 describes existing approaches to in-depth anlysis and description of objects, some of which are used in this work.

2.2 The OAIS model and Preservation Planning

Many repositories follow the Reference Model for an Open Archival Information System (OAIS) described in [ISO03]. The OAIS model was published 2002 by the Consultative Committee for Space Data Systems (CCSDS) and adopted as ISO standard ISO 14721:2003. It has proven to be a very useful high-level reference model, describing functional entities and the exchange of information between them. The OAIS describes an archive, consisting of an organisation and its systems, that has the responsibility for long-term archival and access to information for a designated user community. Because of its growing acceptance in the community, OAIS is the most common framework for digital preservation systems. The specification provides a functional model and an information model. However, the level of abstraction is very high, and the model does not provide concrete guidance.

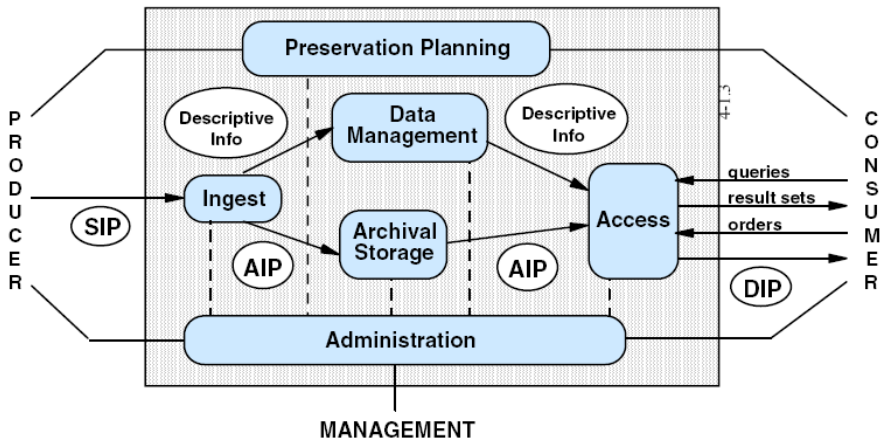


Figure 2.1: The OAIS model [ISO03]

Figure 2.1 shows a high-level overview of the model's main functions and its environment. Producers deposit Submission Information Packages (SIPs) into the archive through the *Ingest* function. Ingest validates each SIP and produces a corresponding Archival Information Package (AIP) which is sent to *Archival Storage*. *Data Management* provides services for creating and maintaining descriptive information and administrative data of the archive. *Access* is responsible for providing Consumers with access to the content of the archive by coordinating queries and producing Dissemination Information Packages (DIPs). Finally, *Preservation Planning* is responsible for monitoring the environment of the archive and providing recommendations for ensuring the long-term availability and accessability of the content. This has to take into account the designated community of users and their needs, but also technical conditions, options and constraints.

Preservation planning has a long history in archival science and conservation, where it traditionally fulfils a slightly different, more high-level role. The Northeast Document Conservation Center defines it as follows:

> Preservation planning is a process by which the general and specific needs for the care of collections are determined, priorities are established, and resources for implementation are identified ... Its main purpose is to define a course of action that will allow an institution to set its present and future preservation agendas [Ogd98, Ogd10].

In the digital domain, preservation planning is one of the core issues addressed by the project Planets[3], which created a distributed service ori-

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

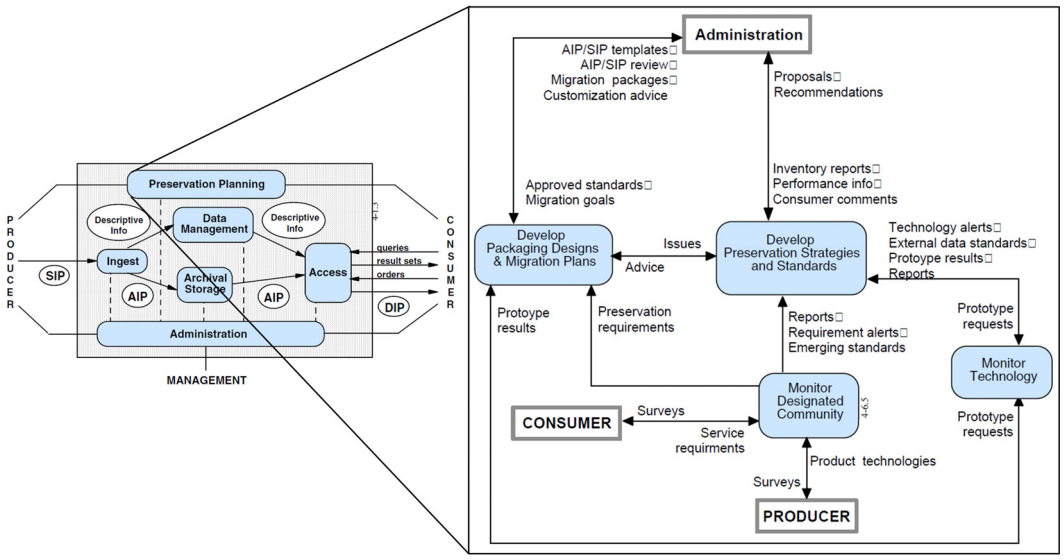


Figure 2.2: Preservation planning in the OAIS model [ISO03]

ented architecture for digital preservation [KSJ+09]. Farquhar presents an overview of the distributed service infrastructure and the main components that form the Planets system [FHY07].

Figure 2.2 shows the main components of the Preservation Planning function in the OAIS model. The *Develop Preservation Strategies and Standards* function is responsible for developing and recommending strategies and standards to preserve the current holdings and new submissions for the future. Its recommendations are provided to the *Develop Packaging Designs and Migration Plans* function as advice to create a detailed migration plan, and to the Administration entity for system evolution. The functions *Monitor Designated Community* and *Monitor Technology* perform a watch that provides reports about developments and changes in the designated community and relevant technologies. *Monitor Technology* offers the functionality to evaluate emerging technologies by prototype requests. The results are first indications for closer consideration of new and untested tools and services.

For example, monitoring the environment may fire a trigger in the event of a format becoming obsolete, thus causing a planning activity to be started in the *Develop Preservation Strategies and Standards* function. This uses input from the other functions to provide a recommendation to *Develop Packaging Designs and Migration Plans*, which defines the preservation plan for the affected set of objects. The continuous monitoring of this plan and the related conditions is then handed over again to the monitoring functions. This ongoing monitoring process on the basis of the monitoring functions is essential for successful continuous preservation management.

The *Manage System Configuration* and the *Consumer Service* function of the *Administration* entity report information about the performance of the archiving system, as well as consumer comments, to the *Develop Preservation Strategies and Standards* function. Consumer comments can imply requirements regarding access, behaviour and usage of the digital objects in the system. Performance information can raise requirements that have to be fulfilled by potential preservation strategies.

The two primary issues thus are the *definition of preservation plans*, including the selection of the most suitable action component as the centerpiece, and *continuous monitoring* of defined plans, the environment, and the repository.

2.3 Trustworthiness in digital repositories

The question of trust as a fundamental issue in digital repositories has received considerable attention [RM06, DSS07, RLG02]. Establishing a trusted and reliable digital archive should increase the confidence of producers and consumers. Producers need to be able to trust in the long-term preservation and accessability of their digital resources held in the repository. On the other side, users need to have confidence in the reliability of the repository and the authenticity of its content.

Rosenthal et al. describe a trusted digital repository as follows.

> A repository is Trusted if it can demonstrate its capacity to fulfil its specified functions, and if those specified functions satisfy an agreed set of minimal criteria which all Trusted Repositories are assumed to require. The requirement that compliance be demonstrable is critical...[4]

Institutions have started to declare their repositories as 'trusted digital repositories' (TDR) or as 'OAIS-compliant'. These claims of trustworthiness or compliance are made easily. However, verifying them objectively is is much more complex.

In recent years, several initiatives have been formed to address this, starting with a joint effort of the Research Library Group (RLG) and the Online Computer Library Center (OCLC) that led to a first definition of attributes and responsibilities of a TDR published 2002 [RLG02]. These initiatives have subsequently produced fundamental principles for trustworthiness and elaborated criteria catalogues that repositories should comply with. They have furthermore developed guidance to set up and analyse TDR operations.

In 2003, RLG and the National Archives and Records Administration founded a joint task force to address digital repository certification. The task

[4] [RBRH+08], p. 9

force developed criteria for long-term reliable digital repositories. In Europe, the Catalogue of Criteria for Trusted Digital Repositories [DSS07], published by the certification working group of NESTOR[5], identifies criteria which facilitate the evaluation of digital repository trustworthiness. Of particular relevance are aspects such as long-term planning, change mechanisms, and the definition of the significant properties of the digital objects that shall be preserved.

A joint declaration of core requirements for a TDR was published by the Digital Curation Centre[6], Digital Preservation Europe[7], NESTOR, and the Center of Research Libraries [Cen07]. The ten fundamental characteristics identified are the following:

1. The repository commits to continuing maintenance of digital objects for identified community/communities.
2. Demonstrates organizational fitness (including financial, staffing structure, and processes) to fulfill its commitment.
3. Acquires and maintains requisite contractual and legal rights and fulfills responsibilities.
4. Has an effective and efficient policy framework.
5. Acquires and ingests digital objects based upon stated criteria that correspond to its commitments and capabilities.
6. Maintains/ensures the integrity, authenticity and usability of digital objects it holds over time.
7. Creates and maintains requisite metadata about actions taken on digital objects during preservation as well as about the relevant production, access support, and usage process contexts before preservation.
8. Fulfills requisite dissemination requirements.
9. Has a strategic program for preservation planning and action.
10. Has technical infrastructure adequate to continuing maintenance and security of its digital objects.[8]

It should be noted that only one out of the ten principles directly refers to purely technical equipment, which illustrates that trustworthiness has to

[5] http://www.langzeitarchivierung.de/eng/index.htm

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

[7] http://www.digitalpreservationeurope.eu/

[8] http://www.crl.edu/archiving-preservation/digital-archives/metrics-assessing-and-certifying/core-re

be achieved on many levels, ranging from organisational issues and strategic planning to information models and usage requirements.

In 2007, the Trustworthy Repositories Audit & Certification: Criteria and Checklist (TRAC) report was published [TO07]. This criteria checklist deals with the organisational and technical infrastructure for trustworthy repositories and covers capabilities of certification for repositories. The declared goal of TRAC is to submit the checklist to ISO for standardisation through a Birds of a Feather group[9]. The next sections will outline some of the relevant criteria, referring to the draft white book published in May 2008 by this standardisation working group [Con08].

2.3.1 Trusted Repositories Audit and Certification criteria

TRAC contains criteria in several aspects that are of specific interest for preservation planning. These include

- Procedures, policies and their evolvement;
- Review and assessment;
- Documented history of changes;
- Transparency and accountability; and
- Monitoring and notification.

From section A, Organisational structure & staffing, Section A3 is of specific relevance, as it describes requirements for a procedural accountability and policy framework. The following aspects are of particular interest.

- A3.1 Repository has defined its designated community(ies) and associated knowledge base(s) and has publicly accessible definitions and policies in place to dictate how its preservation service requirements will be met.
- A3.2 Repository has procedures and policies in place, and mechanisms for their review, update, and development as the repository grows and as technology and community practice evolve.
- A3.4 Repository is committed to formal, periodic review and assessment to ensure responsiveness to technological developments and evolving requirements.
- A3.5 Repository has policies and procedures to ensure that feedback from producers and users is sought and addressed over time.

[9] `http://wiki.digitalrepositoryauditandcertification.org`

- A3.6 Repository has a documented history of the changes to its operations, procedures, software, and hardware.
- A3.7 Repository commits to transparency and accountability in all actions supporting the operation and management of the repository, especially those that affect the preservation of digital content over time.
- A3.8 Repository commits to defining, collecting, tracking, and providing, on demand, its information integrity measurements.

These criteria point to a strong need for clearly defined roles, responsibilities, and procedures, and document that these need to be transparently evolving along the needs of the organisation's operation.

Section *B* of the catalogue covers digital object management responsibilities of a repository. Of particular relevance in this context is criterion B1.1, which requires the repository to be explicit about significant properties and guarantees given for the information content that the repository is responsible for:

- B1.1 Repository identifies properties or information content it will preserve for digital objects.

Section *B3* of the TRAC catalogue explicitly lists aspects considered necessary to prove the trustworthiness of a digital repository with respect to the *preservation planning* function.

- B3.1 Repository has documented preservation strategies.
- B3.2 Repository has mechanisms in place for monitoring and notification when Representation Information (including formats) approaches obsolescence or is no longer viable.
- B3.3 Repository has mechanisms to change its preservation plans as a result of its monitoring activities.
- B3.4 Repository can provide evidence of the effectiveness of its preservation planning.

Thus, the planning function is expected to provide documented and provably effective preservation plans. These must be subjected to monitoring and continuous evolvement, enabling the repository to react to changes in the environment.

2.3.2 NESTOR Criteria

Similar to the TRAC checklist, the Catalogue of Criteria for Trusted Digital Repositories published by the certification working group of NESTOR identifies criteria which facilitate the evaluation of digital repository trustworthiness [DSS07]. The following criteria are particularly relevant to our work:

- 4.4 The digital repository engages in long-term planning.
- 5.3 The digital repository reacts to substantial changes.
- 8. The digital repository has a strategic plan for its technical preservation measures.
- 9.2 The digital repository identifies which characteristics of the digital objects are significant for information preservation.

2.3.3 Repository planning and assessment

The criteria catalogues of TRAC and Nestor have defined essential characteristics that should be fulfilled by repositories in order to be trustworthy. However, they do not provide guidance on how to fulfil these criteria, and they do not directly support repositories in improving their operations according to the requirements. To address these gaps, the European project DPE[10] has developed the two complementary tools PLATTER and DRAMBORA.

PLATTER, the Planning Tool for Trusted Electronic Repositories, is a guiding framework designed to enable repository planners to plan the development of objectives and targets in order to establish trust among the stakeholders. The framework tries to address the diversity of organisations running digital repositories by specifying a process framework and allowing them to define goals that correspond to their situation, instead of prescribing fixed criteria.

The procedure starts by classifying a repository along the axes of purpose or function, scale, operation, and implementation. For example, planning will distinguish between government archives and commercial repositories, and define legal acquisition rights. It will further document the expected amount of archival material and specify the sensitivity of data, modes of access, and corresponding restrictions. Similarly, the classification documents implementation choices such as the storage and software strategies; i.e., whether software is developed in-house or acquired from a commercial vendor, how the system will be supported in the future, etc.

The planning cycle defined by PLATTER is shown in Figure 2.3. Strategic planning clarifies an organisation's mandate and primary stakeholders

[10] http://www.digitalpreservationeurope.eu/

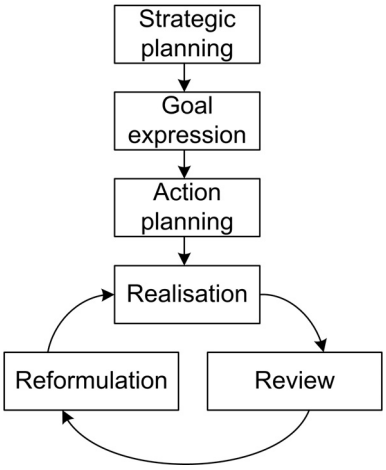


Figure 2.3: The PLATTER Planning Cycle

and provides the foundation of subsequent decisions. Operational planning defines concrete goals in a specific, measurable way, and describes the action path envisioned to fulfil them. Delivering the implementation requires an iterative cycle repeating the related activities of realisation, review, and reformulation, to setup and continually improve repository operation.

PLATTER incorporates the ten fundamental principles described above into the planning process by associating Strategic Objective Plans to the core principles and translating them to realistic and clearly assigned goals. The Strategic Objective Plans thus cover aspects as diverse as business planning and staffing, technical systems, and succession planning.

The tool is 'concerned exclusively with management of the objectives and targets of [a] repository'[11]. The claim is that an organisation using the tool for repository planning will be well prepared for undergoing certification according to upcoming initiatives, since the goals and objectives will be aligned with commonly accepted fundamental principles, and the organisation will possess extensive and exact documentation corresponding to the strategic objectives.

To analyse and assess the operations of an existing repository, an organisation can refer to DRAMBORA, the Digital Repository Audit Method Based on Risk Assessment[12]. DRAMBORA is a risk analysis method that adapts standard risk management models and tailors them to meet the specifics of the repository domain. It is not meant as a certification per se, but instead as a self-assessment tool that organisations can use for analysis and improvement.

[11] [RBRH+08], p. 44

[12] `http://www.repositoryaudit.eu/`

The underlying assumption is that all preservation is ultimately risk management. Through a thorough analysis and documentation of the repository's assets and activities, potential threats can be derived and risks identified. Risk impact can be considered along the lines of damages to assets such as finances, reputation, staff, or the content itself. Ultimately, the loss of digital object authenticity and understandability has to be avoided.

Figure 2.4 shows the main steps of the risk assessment workflow. It consists of four main phases:

1. **Define the audit scope.** – At the outset, the purpose and scope of the audit has to be defined clearly.

2. **Document the context.** – The next two steps formalise the context of subsequent analysis by determining functional classes that can be used to identify and organise activities and assets.

3. **Formalise and document the organisation.** – This extensive documentation phase formalises the organisation's mandate, applying constraints, goals and objectives, and finally activities and assets. For example, organisational assets include information such as databases and contracts, but also software, physical assets, processes, people, and intangibles such as the reputation of the repository. Associated activities as well as assets are linked to individual responsible actors.

4. **Identify and assess risks.** – Starting with activities and assets, the goal is now to identify vulnerabilities and potential threats. For example, assets or activities may fail to achieve relevant goals, and both internal or external threats may pose obstacles to the success of activities. This extensive search stage feeds into an assessment of the corresponding risks in terms of probability and impact. The assessment may also identify relationships between risks. Management of risks can be along the lines of mitigation, avoidance, or acceptance.

Consider one of the fundamental assets of a repository – its mandate. One risk will be that the repository loses its mandate, which would remove the basis for its existence. Further, a substantial change to the mandate could render the activities incompatible with the new mandate – for instance, if the scope of responsibility is changed by new legislation. This risk affects management and administration procedures and has to be avoided. One approach would be to demonstrate effectiveness by getting certification; but in parallel, a succession plan may be needed to ensure that valuable content is not lost entirely, should the risk become manifest.

As another example, preservation actions carry the inherent risk of resulting in loss of information or compromised integrity of content. For example, a repository might convert documents so that the layout is compromised,

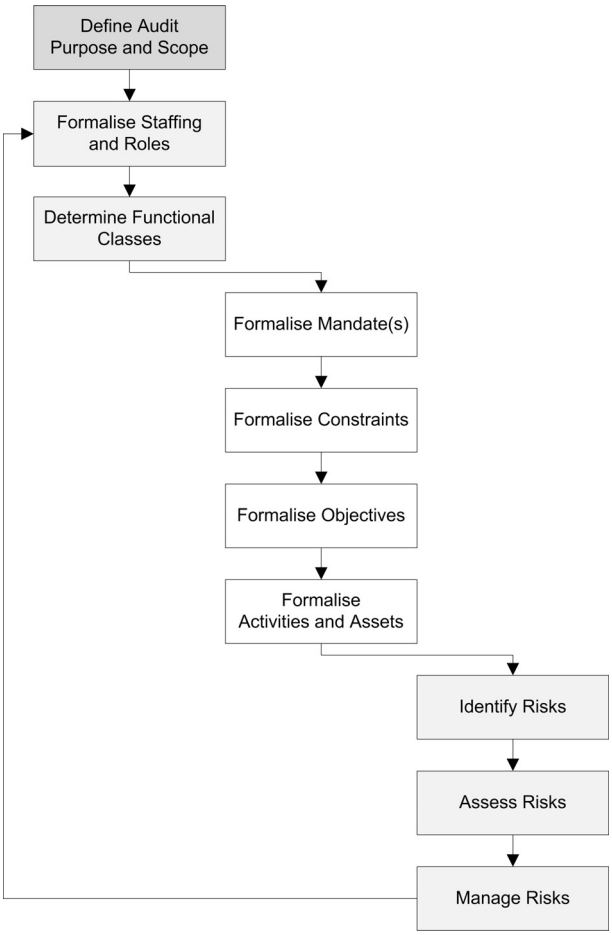


Figure 2.4: The DRAMBORA workflow

but the user community regards this as an essential property. This would severely affect the value of the central asset of the organisation: its content. To avoid this risk, the repository should evaluate strategies prior to execution, and could try to rely on reversible actions in the event of unexpected outcomes. On the other hand, policies can describe acceptable levels of loss that are tolerated in case the risk becomes manifest.

DRAMBORA strongly emphasises flexibility. Instead of prescribing specific risk management policies, auditors should describe their chosen strategy and document performance, targets, and responsibilities, to be able to reassess the success of their measures later.

2.3.4 Synthesis and relationships

The criteria catalogues of TRAC and Nestor set out important considerations that the planning functions and operations of a repository should meet. In contrast to these prescriptive criteria catalogues, the PLATTER toolkit provides a checklist and guidance document for repository planning on a high level. The intention is that repository organisations specify their objectives and targets in groups of strategic objective plans that evolve in cycles. Sharing these enables them to compare their targets and key performance indicators with those of other repositories.

To evaluate the fulfilment of these goals and the risks the repository operations are facing, a repository organisation can use DRAMBORA to conduct a self-assessment by identifying assets, activities and associated risks in a structured way. The method adapts standard risk assessment principles and tailors them to digital repository assessment.

Thus, while the criteria catalogues are geared towards being a checklist for certification, PLATTER and DRAMBORA are guiding repository planners and supporting them with mechanisms for assessment and improvement on a high level. On a more detailed level, we will discuss the relation of the preservation planning approach described in this thesis to the criteria catalogues, and how the contained prescriptive criteria are supported, in Section 3.6.

2.4 Preservation actions

2.4.1 Objects and environments

If a digital object depends on a certain environment to be rendered properly, the most straightforward approach to keeping the object accessible is to keep the environment intact. This is referred to as the **museum approach**. While at first sight the idea seemed appealing, experience quickly showed that it is entirely infeasible for most cases. Technology advancements not only have the effect that spare parts cease to be available soon after a certain family of products drops out of the market. They also imply that future users do not have the knowledge to handle the preserved technology in a satisfying way. Moreover, they make it extremely difficult to extract digital content to reuse it in different environments, such as providing online acccess to old recordings. The museum approach is thus not considered a viable *long-term* strategy, and the two dominant types of preservation actions today are migration and emulation.

Migration requires the repeated copying or conversion of digital objects from one technology to a more stable or current, be it hardware or software. Each migration incurs certain risks and preserves only a certain fraction of the characteristics of a digital object. The Council of Library

and Information Resources (CLIR) described experiences with migration in [LKR+00], where different kinds of risks for a migration project are discussed. Migration is often used to create dissemination packages for user access [MWS02, WB08].

Emulation as the second important strategy strives to reproduce all essential characteristics of the performance of a system, allowing programs and media designed for a particular environment to operate in a different, newer setting. Jeff Rothenberg [Rot99] envisions a framework of an ideal preservation surrounding. The Universal Virtual Computer (UVC) concept [HVDDVEM05] uses elements of both migration and emulation. It simulates a basic architecture including memory, register and rules. In the future only a single emulation layer between the UVC and the computer is necessary to reconstruct a digital object in its original appearance. Recently, Van der Hoeven presented an emerging approach to emulation called *Modular emulation* in [vdHvW05].

2.4.2 Evaluating preservation actions

A growing number of components performing migration and emulation is available today; the question to be answered before putting one of them to use seems simple: Which of the components should we use?

Evaluating preservation strategies has been an issue for several years. The Dutch Testbed, designed by the National Archives of the Netherlands to facilitate experimentation in digital preservation, consisted of a structured experiment workflow and a computer laboratory for comparing the screen renderings of different applications by a visual judgement of adjacent computer monitors. A series of experiments investigated different strategies for preserving text documents, emails, spreadsheets, and databases [Dig02, SV04, Tes01]. Recently, the Planets Testbed has taken up the approach and strives to provide a public platform for performing standardised experiments and sharing experiences [AHJ+08].

Rauch introduced utility analysis as an evaluation metric for digital preservation [RR04, Rau04]. He showed that the hierarchical structure of utility analysis is applicable and well suited to the evaluation problem.

The DELOS[13] Testbed combines these approaches. It focuses on the elicitation and documentation of objectives, as well as running experiments and evaluating experiments in a structured way [SRR+06]. Strodl et. al. [SBNR07] further develop the evaluation method. Figure 2.5, reproduced from [SRR+06], shows the 14-step workflow of the DELOS Testbed. It is grouped in three phases:

1. Define requirements,
2. Evaluate alternatives, and

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

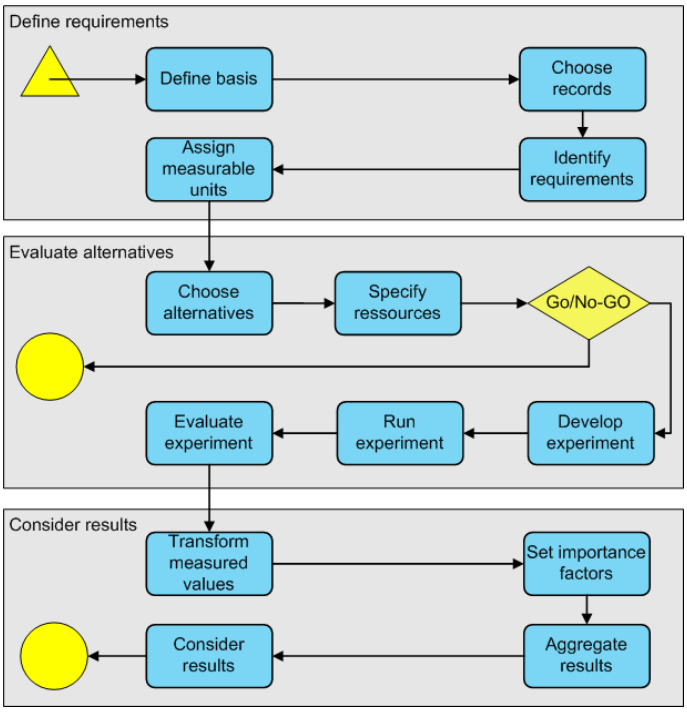


Figure 2.5: DELOS Testbed workflow

3. Consider results.

Requirements are defined in a tree structure, based on utility analysis. Experiments are carried out on sample objects taken from the collection. Manual inspection is used to evaluate each considered action along the requirements defined in the so-called 'objective tree'. A detailed description of the steps is presented in [SRR+06].

The approach has since then been evaluated in a series of case studies [SBNR07, BKKR07, BSN+07]. The work described in this thesis builds on the evaluation method and extends it to address the issues of plan specification and monitoring. It further presents extensive tool support and introduces controlled experimentation and automated measurements as primary means of data collection for evaluation and monitoring. We will describe our extended methodology in Chapter 3.

2.4.3 An early case study: Electronic documents

The scenario

The Austrian National Library (ONB) has the obligation to collect and preserve master theses from Austrian universities. The theses are provided

to the library in PDF format. The ONB provides guidelines for creating preservable PDFs [Hor05], but at the moment the library is not able to legally enforce these guidelines. This case study, originally published in [SBN+07, BSN+07], gave a starting point to identify the requirements and goals for the digital preservation of master theses. It furthermore allowed a first evaluation of the various preservation actions being considered, and enabled an informed critical look on the evaluation approach developed in DELOS.

Evaluation

In a brainstorming workshop the requirements for this specific application area were collected. The resulting objective tree shows a strong focus on the structure, content and appearance of the objects; especially layout and structure of the documents need to be preserved. Characteristics concerning object structure include among others

- Document structure (chapters, sections),
- Reference tables (table of content, list of figures)
- Line and page breaks,
- Headers and footers,
- Footnotes,
- Equations (size, position, structure, caption),
- Figures (size, position, structure, caption), and
- Tables (size, position, structure, caption)

The elicitation and definition of requirements during a brainstorming session, as well as the subsequent structuring to form an objective tree, were performed in a traditional manner, using staples of post-it notes on a white-board. The tree was then documented in the DELOS Testbed software.

The following migration solutions were evaluated.

1. Conversion to plain-text format using Adobe Acrobat 7 Professional;
2. Conversion to Rich Text Format (RTF) using SoftInterface ConvertDoc 3.82;
3. Conversion to RTF using Adobe Acrobat 7 Professional;
4. Conversion to Multipage TIFF using Universal Document Converter 4.1;

5. Conversion to PDF/A using Adobe Acrobat 7 Professional;[14]
6. Conversion to lossless JPEG2000 using Adobe Acrobat 7 Professional;
7. Conversion to Encapsulated PostScript (EPS) using Adobe Acrobat 7 Professional;
8. We also evaluated the alternative of not migrating at all, which leaves the documents in their original formats. As there are multiple versions of PDF, this of course incurs additional risks.

All experiments were executed on Windows XP professional on a sample set of five master's theses from the Vienna University of Technology.

The results show that the migration to PDF/A using Adobe Acrobat 7 Professional ranks on top, followed by migration to the formats TIFF, EPS and JPEG2000; far behind are conversion to RTF and plain text.

The alternative PDF/A basically preserves all core document characteristics in a widely adopted file format and shows good migration process performance.

While the option of leaving the documents in their original PDF format(s) performs well with respect to most criteria, some essential requirements are not met. These are the deactivation of scripting and security mechanisms, which are regarded a knock-out criterion that must be fulfilled. The alternatives TIFF, EPS and JPEG show very good appearance, but have weaknesses regarding criteria such as 'content machine readable'. Furthermore, as the migration to JPEG and EPS produces one output file for each page, the object coherence is not as well preserved as in a PDF/A document. Both RTF solutions exhibit major weaknesses in appearance and structure of the documents, specifically with respect to tables and equations as well as character encoding and line breaks. Object characteristics show a clear advantage for ConvertDoc, which was able to preserve the layout of headers and footers as opposed to Adobe Acrobat. Still, costs and the technical advantages of the Acrobat tool, such as macro support and customisation, compensate for this difference and lead to an equal score.

The loss of essential characteristics means that the plain text format fails to fulfil a number of minimum requirements regarding the preservation of important artefacts like tables and figures as well as appearance characteristics like font types and sizes. Multimedia content proved to be a difficult task: None of the tested alternatives was able to preserve embedded audio and video content. This issue could be solved in two ways:

1. Use a tool for automated extraction of multimedia content from PDF.

[14] (Note that the generated PDF/A is not completely consistent with the PDF/A-ISO-Standard [4])

2. Solve the problem on an organisational level by issuing a submission policy which states that multimedia objects have to be provided separately.

In both cases, a separate preservation strategy for the multimedia content has to be devised. Depending on whether preserving multimedia content is a primary goal to be fulfilled, the final recommendation resulting from the evaluation of the experiments is to

1. use migration to PDF/A with Adobe Acrobat 7 Professional or
2. combine the alternative PDF/A with a multimedia extraction tool or a submission policy.

2.4.4 Observations

The question of evalating and selecting the optimal component performing a preservation action proves rather complicated for a number of reasons.

- **Varying quality across tools.** – While the functional attributes of preservation action components are very homogeneous, the non-functional properties are not. Each tool has very particular strengths and weaknesses. Some migration tools are unable to convert tables properly; others show weaknesses in converting character encoding. With emulation environments, support for specific features varies, and so does performance. A migration tool that works well on one type of input does not necessarily perform adequately on a different input format or deliver a satisfactory transformation into a different output format. At the same time, the authenticity of objects and the integrity of information presented to the user is a most fundamental requirement for any repository, as we have seen illustrated in the previous sections. The declared capability of a tool is thus only a first indicator of suitability.
- **Varying properties across content.** – Even within seemingly homogeneous and simple types of content, such as scanned images, there is often a vast variety of properties to be found. For example, the exact features of scanned images will depend on the scanning equipment and the workflow software that was used to embed or deposit the colour profiles; and common office documents exhibit a surprising variety of complex features that range from embedded tables to active content, encryption, dynamic fonts, or software that is contained in documents. How each of these properties is handled by any of the action components that are available cannot be simply deducted from feature tables, but often has to be analysed in detail in empirical studies.

- **Varying usage across communities.** – Different users with different equipment will show differences in the ways they intend to access and use certain content. This means that the very same quality of a certain tool, having the same known or unknown effect on a certain object, may be perceived as perfectly acceptable by one designated community, while considered intolerable by another. For example, when converting a collection of documents for online access, the loss of line breaks might be perfectly acceptable in one case. But if the user community is used to referring to line numbers in order to locate certain quotes or mark phrases in manuscripts, the loss of this property is ruining an access feature they may regard as essential.

- **Varying requirements across scenarios.** – The choice of component cannot just be based on the shared experience of other cases, but also has to take into account the specifics of the access scenarios. These specifics may even vary within a community. Different collections will be accessed in varying ways by a certain user group, taking into account their interest, but also their pecularities. For example, one of a certain number of scanned books may contain miniature scripts that require very high resolution access copies. The concrete scenario of delivery and access to content may have an impact on the desired properties of content as well as on necessary non-functional properties such as the speed of access. This may for example prohibit the use of on-demand migration, rendering environments or emulators for performance reasons.

- **Varying risk tolerance across collections.** – Even considering one organisation and one designated community, different tolerance levels may apply to certain collections. Valuable and rare objects will be given priority and risk tolerance on the side of the organisation will be low, leading to a higher availability of resources for preservation.

- **Varying preferences and constraints across organisations.** – If requirements vary within an organisation according to different content profiles, user communities, and scenarios, the diversity becomes only more complex when considering the differences between organisations. Not only are organisations different from each other and embedded in diverse legal frameworks and environments; many organisations also do not have clearly articulated these constraints, so that it is hard to draw conclusions and build analogies between different approaches and component choices.

- **Varying costs and compatibility across environments.** – Costs depend on various factors, of which the licensing fees of a certain component form only a small fraction. Depending on data volumes, storage

architectures and policies, scalability may be a strong concern; technical compatibility to existing IT infrastructure will further constrain the choice of potential components.

- **Shifting constraints, priorities, and requirements.** – All of the abovementioned difficulties are subject to constant changes. Legislation is altered, user communities drift, and priorities shift. These changes have to be taken into account in a decision framework, as once seemingly optimal decisions may turn out to be in need for revision in the near or far future.

Each tool has very particular strengths and weaknesses, and most often, there is no optimal solution. On the other hand, requirements vary across institutions and domains, and for each setting, very specific constraints apply that need to be considered. The decision for a component is further complicated by the variation in the digital content that has to be preserved. The selection of the most suitable component to keep a type of digital object alive when the original technical environment ceases to exist is thus a highly complex selection problem with several peculiarities: Highly homogeneous and well-specified core functionality across components, complex evaluation of quality across settings, and a high need for automation, standardisation, and documentation.

The components need to be monitored continually and will likely be replaced in the future when the requirements or characteristics of source objects have shifted or the alternative components have improved so that preferences have changed as an effect. Thus, it is not a one-off component selection problem but a recurring issue, where continuous monitoring is needed to keep track of the performance of deployed components. Deviation from specified monitoring conditions leads to an identified need for re-evaluation and a potential revision of the selection at a later point in time [BR10].

From the experience in the early case study, several conclusions were drawn.

- Evaluating preservation strategies is a multi-objective decision making problem with various stakeholders involved. The structured approach on the basis of utility analysis is in principle well suited for the problem.

- The effort needed to manually construct and evaluate the tree structure is significant. Tool support is needed for all steps to automate the procedures of requirements and evaluation.

- The result of the method is a recommendation for one of the considered actions. However, it is left unclear how to proceed with this recommendation. The question of monitoring is not covered.

- Decision makers have difficulties in exactly defining their constraints and influence factors, and are unable to trace changes in these influence factors to the outcomes of decisions.

- While the utility analysis approach is practical, there is a significant body of work on multi-objective decision making and in the area of component selection in software engineering that has not been considered. The next section outlines some of this work and draws conclusions.

2.5 Component evaluation and selection

2.5.1 Overview

The selection problem in digital preservation is a domain-specific instance of the general problem of component evaluation and selection [Rol99]. This problem of selecting the right software component out of a range of choices to fit in a larger system, often referred to as component selection or Commercial-off-the-shelf (COTS) selection, has received considerable attention in the field of software engineering during the last two decades [JS09]. The principal problem appears in a wide range of different scenarios, from component based software development (CBSD) to web service selection and composition.

Most approaches to component evaluation and selection apply a goal-driven approach [vL01] to support the selection of the most suitable software component. Product quality models and domain-specific criteria catalogues are being developed to structure and reuse knowledge and experience. Most selection methods conform to a general component selection process with the steps *Define criteria, Search for products, Create shortlist, Evaluate candidates, Analyze data and select product* [MRE07]. In general, these approaches are geared towards flexibility and applicability in a wide range of domains, and assume that component selection is a one-off procedure carried out within a development effort to build a new system. In these scenarios, evaluation of candidate components against requirements can be done in a largely manual way, which usually implies, but also allows for, high levels of complexity.

Thorough reviews of literature have been presented in [JS09, MRE07]. One of the first selection methods presented was the Off-the-Shelf-Option (OTSO) [Kon95, Kon96]. It provides a repeatable process for evaluating, selecting and implementing reusable software components. OTSO relies on the Analytic Hierarchy Process (AHP) [Saa90] to facilitate evaluation against hierarchically defined criteria. Using AHP, the relative importance factor of each criterion is obtained through series of pairwise comparisons which result in a ranking matrix. The resulting eigenvalues are used for calculating relative weights of all factors on each level of the hierarchy.

2.5.2 Multi criteria decision making

Component selection is one of many cases of decision making under multiple competing objectives [KR93]. Ncube discusses limitations of multi-criteria decision making techniques such as the Weighted Scoring Method (WSM) or Analytic Hierarchy Process (AHP) , which are often used in conmponent selection [ND02]. While WSM has earned criticism for the necessity to determine criteria weights in advance without having seen alternative solutions, AHP is problematic because of the sheer complexity and effort that is introduced by the pairwise comparison of criteria [NS07b, MN98, ND02]. For n criteria, the number of comparisons is $n(n-1)/2$. Perini describes results of an empirical study comparing AHP with a Case-based ranking approach called *CBRank* that reduces the number of pairwise comparisons. They analyse time consumption, ease of use, and accuracy. AHP outperforned CBRank in terms of accuracy, but was far more time-consuming to use [PRS09]. Another drawback of AHP is that adding or deleting candidate components can lead to changes in the ranking of the other candidates [JS09]. For example, consider a candidate ranking of three components *A, B, C*. Adding a candidate *D* that performs worse than the three others can in fact lead to a change within the ranking of the existing components, so that the final ranking might be *C, A, B, D*. This violates the *Independence of irrelevant alternatives* (IIA) property, which since long is a subject of discussion in social choice models and econometrics [Deb60, Ray73, McL96]. The IIA property can be very useful for cases where a decision maker only wants to evaluate a subset of the choices instead of all possible alternatives, since it allows the reduction of the set of choices and exclusion of the irrelevant alternatives. Its violation can be a significant disruption to consistent decision making.

Alves discusses several peculiarities of COTS requirements engineering in [Alv03], introducing the notion of conflict management. Goals and features need to be acquired and matched, and arising conflicts resolved. They propose a selection process called CRE which identifies four dimensions of COTS selection: domain coverage, time restriction, costs rating, and vendor guarantees [AC01]. While CRE emphasises non-functional requirements, it "does not address the issues of quality testing" [Alv03]. The PORE method [NM99] progressively filters candidates by iteratively refined evaluation criteria.

2.5.3 Quality models

Considerable effort has been spent in standardising software product quality models. The ISO/IEC 9126 standards [ISO01] provide guidance for quality models and define a hierarchy of high-level quality attributes. Quality measures are based on measurement procedures recommended in ISO 15939 [ISO07b]. The model distinguishes between three types of quality:

1. *Internal software quality* refers to static attributes such as complexity measures that can be obtained by analysing the source code. Measures are primarily made during the development stages.

2. *External software quality* attributes refer to the behaviour of a system that includes the component such as the number of failures found during testing. Measurements are generally taken during testing and operation.

3. *Quality in use* describes inhowfar the usage of a component satisfies user needs. Measurements of these attributes have to be taken in a realistic environment [ISO07a].

The successor ISO 25000 standards for Software Product Quality Requirements and Evaluation (SQuaRE) combine the ISO 9126 models with evaluation procedures based on ISO 14598 [ISO99]. They also define requirements on the specification of software product quality criteria [ISO07a].

Franch describes hierachical quality models for component selection based on the ISO 9126 quality model in [FC03]. They propose a six-step method for defining a hierarchy of quality attributes for a specific domain in a top-down fashion:

1. Determining quality subcharacteristics,

2. Defining a hierarchy of subcharacteristics,

3. Decomposing subcharacteristics into attributes,

4. Decomposing derived attributes into basic ones,

5. Stating relationships between quality entities, and

6. Determining metrics for attributes.

This procedure of hierarchical structuring is applicable in the method described in this thesis which relies on a similar tree structure.

Carvallo discusses experiences with quality criteria [CFQ07] and proposes a method called RECSS to support structuring of the system environment and requirements elicitation [CFQ08]. The approach results in thoroughly defined hierarchies of quality factors and the relationships between them. However, requirements are not discussed in detail. They focus on creating a match between features and user requirements, emphasising requirements flexibility.

2.5.4 Approaches to selection and measurement

Techniques from Search-Based Software Engineering [HJ01] have increasingly been applied to the problem of component selection, primarily to rank and select components and find near-optimal solutions for the selection problem in large search spaces. Baker describes ranking and selection as a feature set selection problem and shows that expert judgement in selecting sets of components from a database is outperformed by automated search algorithms [BHSS06]. Similarly, Vijayalakshmi et. al. apply a genetic algorithm with a WSM fitness function to select components from a large set of candidates [VRA08]. DEER [CCMP08] focuses on the tasks of ranking and selection of components in the requirements stage of component-based development. It assumes that evaluation values are known and provides a heuristic to rank and select components or assemblies of multiple components.

Ochs et. al. describe a method for measurement-based assessment and selection called CAP [OPCDNK01, OPCDNK00]. The method comprises a four-level taxonomy of evaluation criteria, several specification and assessment activities, and a control workflow. Like other methods, CAP is based on the view that measurement of all applicable criteria is too difficult and expensive, which is true for many classic CBSD scenarios. Hence, the CAP heuristic was designed to increase efficiency by reducing the number of actual measurement operations. CAP goes considerably further than other methods in defining accurate metrics. Yet, it does not appear to use controlled experimentation and automated measurements to collect data on the components, even though this may be both feasible and desirable in other settings. The primary goal is to avoid taking measurements to increase the efficiency of the decision making procedure.

Kitchenham et. al. discuss software measurement in [KHL01] and present a software data model and several rules for data collection. They state that in order to ensure repeatability and comparability, the counting rules for each measure are essential, since measurement conditions and procedures have a strong impact on the comparability of values. Yet, software measurement is still a relatively young discipline, and terms are in flux [GBC+06]. When discussing measurement concepts in this thesis, we choose to rely on terms generally understood in the digital preservation domain, such as significant properties [DF09], as opposed to the still inconsistent literature of software measurement.

2.5.5 Components, services, and trust

Work on component selection arose from the component-based system development paradigm [CPV03], where components can be selected for inclusion into a system that is being built during any phase from requirements analysis

	Component selection for CBSD	**Web service selection and composition**
Scenario	Static selection sometimes followed by monitoring	Dynamic selection and continuos monitoring of QoS
Granularity of components	Different levels of granularity and features	Fine-grained services with very homogeneous features
Quality attributes	Focus on feature sets and quality models	Focus on low-overhead measurements of generic attributes in production
Data collection	Expert knowledge and standardised quality models	Automated measurements with minimum overhead in service delivery
Target values	Structured definition of requirements and desired target ranges	Automated optimisation of basic QoS attributes

Table 2.1: Range of approaches to component evaluation and selection

to system integration. The level of requirements and available information throughout these phases generally changes, and different methods have been suggested for different stages. One of the primary requirements for assessment methods in that context is efficiency. Once the components are selected and the system is built, there is often no reassessment. However, Yang et.al. discuss the issues of changing COTS components in large component-based systems and emphasise the need for monitoring and reassessing components, pointing out that typical release cycles average about 10 months [YBBP05].

Towards the other end of the dynamics spectrum we find web service composition and web service quality (QoS) [Men02, DS05, Ran03, EMT07, TGRS04]. Web services are often selected and composed dynamically at runtime, and QoS information has to be collected on-the-fly with little overhead. Quality attributes need to be modelled, measured, evaluated, and monitored constantly. Trust management is considered a central aspect of current web service research [BKL+09, MS04, SMNBC09].

Table 2.1 highlights some key aspects that differ across the range of component selection scenarios. While CBSD generally focuses on a static selection scenario where the features of components of varying granularity are evaluated manually according to standardised quality catalogues, web service selection focuses on dynamic selection of fine-grained components using automated measurements of generic attributes. The approach we are presenting here draws from both ends of the spectrum; we will discuss the relation to existing methods in Section 3.3.6.

2.6 Systematic characterisation

2.6.1 Overview

While migration and emulation perform the primary action functionality of rendering or converting objects, *characterisation* tools are needed to analyse and describe the content and structure of the digital objects we need to preserve. The evaluation of action components for digital preservation has to rely on an analysis of the logical structure and the content of the objects in order to analyse which aspects have been preserved by a specific preservation action.

Converting any number of documents from one format into another, i.e. transforming the actual representation of their content, inevitably raises the issue of preserving authenticity. Moreover, to confidently choose between alternative target formats and tools, one has to evaluate their suitability in a given context. This leads to the following underlying questions.

1. Which information contained in the old format is also contained in the new format?
2. Which information relevant to the usage of the content of the old format is contained in the new format?
3. Is the conversion process *a(old,new)* better then *b(old,new)*, i.e. does it preserve more of the relevant information contained within the object?

A number of tools and services have been developed that perform content characterisation specifically for digital preservation. The National Library of New Zealand Metadata Extraction Tool[15] extracts preservation metadata for various input file formats. The Digital Repository Object Identification tool (DROID)[16] and JHOVE [17] perform file format identification, validation, and characterisation of digital objects. However, both tools only extract metadata such as the presence of specific file format features in a document; they do not describe the content of a document. The newer tool FITS[18] (File Information Toolset) acts as a unifying wrapper for a number of tools including DROID and JHOVE. In contrast, the extensible characterisation languages [BRH+08a] described below perform in-depth characterisation and strive to extract the complete informational content of digital objects.

Some solutions exist for transforming, matching, and comparing structured documents. Díaz describes the usage of XML for handling the conversion problems that arise in the exchange of business documents between organisations [DWB02]. Canfield presents an algorithm for approximate XML document matching in [CX05].

[15] `http://meta-extractor.sourceforge.net/`
[16] `http://droid.sourceforge.net`
[17] `http://hul.harvard.edu/jhove`
[18] `http://code.google.com/p/fits/`

In the area of grid computing, the Open Grid Forum Data Format Description Language Working Group has been working on a language called DFDL [BW04, Pow10] which extends XML Schema. The aim is to describe *the structure of binary and character encoded (ASCII/Unicode) files and data streams so that their format, structure, and metadata can be exposed*[19]. Thus the DFDL creates a mapping between formatted files and corresponding XML representations. The PADS language, on the other hand, is a domain-specific language based on C-structures that aims at performance-oriented processing of large, simple structured files [FG05], while the Extensible Scientific Interchange Language (XSIL) focuses on scientific data objects [BLPW99].

The following sections describe the eXtensible Characterisation Languages (XCL) that support the automated validation of document conversions and the evaluation of migration quality by hierarchically decomposing a document and representing documents from different sources in an abstract XML language. The description language XCDL provides an abstract representation of digital content in XML, while the extraction language XCEL allows an extraction engine to create such an abstract description by mapping file format structures to XCDL concepts. We present the context of the development of these languages and tools and describe the overall concept and features of the languages. We further give examples and show how the languages can be applied to the evaluation of digital preservation solutions in the context of preservation planning.

The XCL languages and tools have been developed by the University at Cologne in the course of the Planets project [20]. Our contribution is in the design of their usage to make them fit for purpose within the evaluation and validation framework. The following sections are thus based on our joint presentation of the XCL framework as published in [BRH+08a, BRH+08b].

2.6.2 The extensible characterisation languages

Comparing information in two different file formats implies the following requisites.

1. An abstract way of expressing the information in a format-neutral model. This is henceforth called an *extensible characterisation definition language (XCDL)*. Such a language should be defined so generic that it supports the description of arbitrary file formats and thus the extraction of characteristics from any given file.

2. A way of extracting information in specific file formats and describing it using XCDL. It would be theoretically possible to create an extraction

[19] `http://forge.ggf.org/sf/projects/dfdl-wg`
[20] `http://planetarium.hki.uni-koeln.de/`

tool for every given file format. An alternative approach is to define a comprehensive *extensible characterisation extraction language (XCEL)* and implement an extractor component that is able to interpret this language.

3. Algorithms for comparing two XCDL descriptions for degrees of equality.

The eXtensible characterisation languages consist of two languages: the description language XCDL and the exctraction language XCEL. The specifications for both languages are publicly available[21].

The Extensible Characterisation Description language (XCDL) allows the representation of characteristics extracted from files. The definition of characteristics is taken in the broadest possible way. Conceptually, an XCDL representation of the information contained within a file can be a complete interpretation of all the information contained in that file.

An XCDL document describes the content of a specific file with format type X, tagged in XML according to the XCDL language specifications, and is processible through an XCDL interpreter. An XCEL document describes what information can be extracted from any given file of format X, enabling an XCEL processor to extract this information and express it in XCDL. XCEL thus creates a mapping between the declarative description of the information in a physical file and its abstract interpretation outside of a format specification. Both XCDL and XCEL are meta-languages defined using XML Schema. In contrast to other applications of XML in digital preservation, XCL does not migrate digital objects as a whole to XML nor store exclusively preservation metadata; it transforms the entire content of an object into an abstract unified form. A key application is the comparison of different representations of the same object in order to validate migration within the preservation planning procedure.

The next sections will describe the basic architecture of both characterisation languages. We will then outline an example of how they can be applied in practice.

The extraction language XCEL

The eXtensible Characterisation Extraction Language (XCEL) is a file format description language for describing file structures in order to allow their parsing by generalised software. The main goal of the XCEL is therefore to provide all tools necessary for describing real-life file formats like PNG, TIFF, PDF or DOCX.[22] The XCEL is a declarative, descriptive, XML-based

[21] `http://planetarium.hki.uni-koeln.de/public/XCL`

[22] The PNG specification is available at `http://www.w3.org/TR/PNG/`.
The TIFF specification is available at `http://partners.adobe.com/public/developer/tiff/index.html`.

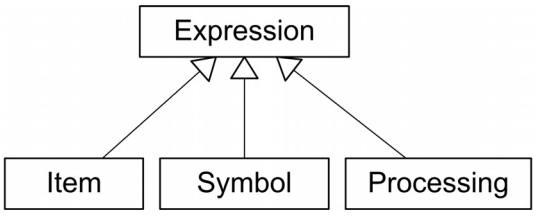


Figure 2.6: The structuring elements of XCEL

language that provides well defined mechanisms for extending certain parts of the language. As an Extraction Language the XCEL has some similarities to other domain specific languages for describing file formats [FMW06]. The Data Format Description Language (DFDL) has a number of common properties with the XCEL; however, there are significant differences. DFDL focuses on scientific data, while XCEL is primarily targeted at file formats typically held in libraries and archives. The DFDL is implemented as an extension of XML-Schema, while the XCEL is a completely new language; the syntax of XCEL can be described with the XML-Schema language. Other approaches like PADS are focusing on the processing of simple but large-scale data formats [FG05]. The distinct goal of XCEL is not extract purely technical entities, such as 'a 3 x 256 array of one byte numbers', but rather characteristics with a semantic meaning, such as 'a colour lookup table'.

An XCEL document comprises the following parts.

1. **Preprocessing** information includes configuration tasks affecting the behaviour of the XCEL interpreter.

2. The **format description** is the core part defining the structure of a file.

3. **Templates** describe recurring structures such as number formats in ASCII based file formats.

4. **Postprocessing** instructions define actions to be performed on the result of the format processing.

Figure 2.6 shows the abstract relations of XCEL expressions. This structure – which seems to be simpler than the one proposed by the DFDL[23] – is based upon the assumption that any file format can be expressed as a set of

The PDF specification is available at `http://www.adobe.com/devnet/pdf/pdf_reference.html`.
The DOCX specification is available at `http://www.ecmainternational.org/news/TC45_current_work/TC45_available_docs.htm`

[23] `http://forge.ggf.org/sf/projects/dfdl-wg`

```
<!-- The complete IDAT chunk is expressed as one item that
     prescribes all its children to appear in the given order -->
<item xsi:type="structuringItem" identifier="IDAT" multiple="true"
optional="true">
  <symbol identifier="IDATLength" interpretation="uint32" length="4"/>
  <symbol identifier="IDATChunkType" interpretation="ASCII"
          value="IDAT" />
  <!-- Set the length of the expression "IDATChunkData"
     to the value given by the expression "IDATLength"-->
  <processing type="pushXCEL" xcelRef="IDATChunkData">
      <processingMethod name="setLength">
          <param valueRef="IDATLength"/>
      </processingMethod>
  </processing>
  <symbol identifier="IDATChunkData" interpretation="uint8"
          name="normData"/>
  <symbol name="IDATCRC" length="4" />
</item>
```

Figure 2.7: XCEL description of a PNG chunk

hierarchies of blocks of content, all of which can be addressed from within but also outside of these hierarchies.

An XCEL format description starts with an *Item*, a container element that can have different content models, similar to the XML-Schema content models. A *Symbol* is an expression that adds a name or ID to a specific part of the byte stream. The *Processing* element models an expression that is used to call built-in methods for the extraction processor. This structure describes file formats in a tree where each branch describes one possible variation. It is the job of the XCEL processor to find out which branch matches to a given file.

Figure 2.7 shows the XCEL description of the IDAT chunk of a PNG image [ISO04b]. Every chunk in PNG consists of the consecutive parts `length`, `chunk type`, `chunk data` and `CRC`. The `length` is a four byte unsigned integer that contains the length of the `chunk data` field, `chunk type` is a four byte ASCII keyword, `chunk data` is a field that can contain any data structure, and `CRC` contains a checksum.

The XCEL processing software ('Extractor') processes the binary files of given formats using the specific XCEL documents created for these formats. Currently there exist XCEL documents for various file formats, focusing on the image, text and audio data domain (e.g. TIFF, PNG, GIF, WAV, and PDF). The Extractor is conceived in such a way as to be able to process any additionally created XCEL document without modifications of its core implementation. Thus, to enlarge the spectrum of supported file formats one only has to write an XCEL document for that format.

The description language XCDL

The result of extracting content from a file using an XCEL document as input for an extractor is a description of the informational content of this file in the description language XCDL. A simple example is provided in Figure 2.8, which provides a part of an XCDL description of a text document containing the phrase 'An **important** word'.

A common characteristic of content models is a separation between primary information and properties of that information. Within the textual domain this separation consists e.g. in the difference between the string 'An important word' and the means by which we indicate that the single words in that string are expressed in specific fonts. The corresponding XCDL representation is provided below. The *normData* tag wraps the primary information in a context-free manner, removing or transforming all format-specific information as well as its specific representation. Text encoding is translated into UTF8 by default. The fonts are described within the *property* tags. In this case we have a property describing the fonts used. For each different font a value set is created. The font name appears as a labelled value referring to exactly defined terms in the XCL properties ontology. The *dataRef* tags define positions within the normalised data by referencing a *propertySet* which indicates where the specific fonts are applied. The *propertySet* furthermore contains references to all related *valueSet* entries, thus creating a bi-directional mapping. This basic structure of separating data and associated properties is common for all underlying content models: In the case of images, e.g., the primary stream of bytes describing the pixels can have properties which are applicable to an image as a whole (e.g. a gamma correction) or only to parts thereof, as for example an embedded explanatory text in the image.

For preservation purposes, the properties extracted from a file may include a category of properties which are not needed to model the content of the file. Consider, e.g., a file format which compresses a part of the data it contains. A proper XCEL extractor, which extracts the content of the file and expresses it in XCDL, has to be able to apply that algorithm in order to decompress the content. Once this is done, the algorithm applied to the original file becomes irrelevant for comparison purposes, as the content is simply the result of its application. For preservation purposes - basically tracking the history of a file and its authenticity - properties like 'originally compressed by algorithm X' can be expressed.

Comparing digital objects

The XCL languages have been designed with the primary goal of automating the validation of content in converted representations within the preservation planning procedure. Figure 2.9 shows a corresponding scenario for applying

```
<object id="o1" >
  <normData type="text" id="nd1">An important word</normData>
  <property id="p94" source="raw" cat="descr" >
      <name id="id86" >pdfBaseFont</name>
      <valueSet id="i_i1_i49_i2_i3" >
          <labValue>
              <val>NimbusRomanNo9L-Regu</val>
              <type>string</type>
          </labValue>
          <dataRef ind="normSpecific" propertySetId="id_0" />
      </valueSet>
      <valueSet id="i_i1_i49_i2_i4" >
          <labValue>
              <val>NimbusRomNo9L-Medi</val>
              <type>string</type>
          </labValue>
          <dataRef ind="normSpecific" propertySetId="id_1" />
      </valueSet>
  </property>
  <property id="p106" source="raw" cat="descr" >
      <name id="id158" >fontSize</name>
      <valueSet id="i_i1_i70_i2" >
          <labValue>
              <val>12</val>
              <type>rational</type>
          </labValue>
          <dataRef ind="normSpecific" propertySetId="id_0" />
      </valueSet>
  </property>
  <propertySet id="id_0" >
      <valueSetRelations>
          <ref valueSetId="i_i1_i49_i2_i3" name="pdfBaseFont" />
          <ref valueSetId="i_i1_i70_i2" name="fontSize" />
      </valueSetRelations>
      <dataRef>
          <ref begin="0" end="1" id="nd1" />
          <ref begin="13" end="16" id="nd1" />
      </dataRef>
  </propertySet>
  <propertySet id="id_1" >
      <valueSetRelations>
          <ref valueSetId="i_i1_i49_i2_i4" name="pdfBaseFont" />
          <ref valueSetId="i_i1_i70_i2" name="fontSize" />
      </valueSetRelations>
      <dataRef>
          <ref begin="2" end="12" id="nd1" />
      </dataRef>
  </propertySet>
</object>
```

Figure 2.8: XCDL representation of primary information and corresponding properties, connected by property sets

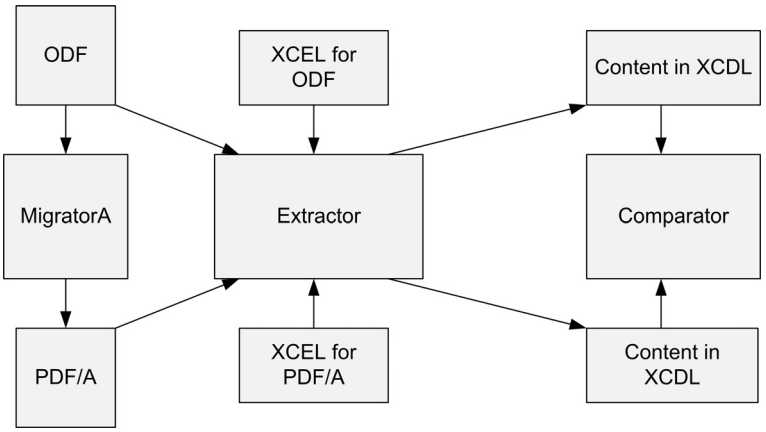


Figure 2.9: Using XCL to compare migrated documents

XCL in the context of format migration. After converting a document from ODF to PDF/A, the XCDL documents of the original and the transformed object can be compared using a comparison tool ('Comparator') for XCDL documents. The property-specific definition of metrics and their implementations as algorithms allow the comparator to identify degrees of equality between two XCDL documents. In its core functionality it loads two XCDL documents, extracts the property sequences and compares them according to comparison metrics which are defined with respect to the types of the values in the value sets. In the example of Figure 2.8, the comparator looks up the defined metrics for property 'Fontname' and executes the comparison according to the metrics definition. This can be a simple binary comparison that checks the XCL ontology for the entries 'Times-Roman' and 'Times-Bold'. For other properties such as as possible deviation of font size, absolute or relative difference measures can be used. This evaluation of a migration process' quality can provide considerable support for the selection of components.

To verify the approach, a benchmark corpus of PNG files[24] was migrated to TIFF. In contrast to other tools such as JHOVE or tiffInfo[25], XCL was able to extract file properties as well as the normalised content from all files. Comparing the *normData* with a tool revealed that conversion of images with certain characteristics had not been successful. For example, some images that contained transparency specifications had been converted incorrectly. In contrast to purely metadata-oriented comparisons, the in-depth characterisation extracting the full content model revealed these issues properly.

For evaluating preservation strategies, preservation planning activities

[24] `http://www.schaik.com/pngsuite`
[25] `http://remotesensing.org/libtiff/man/tiffinfo.1.html`

define requirements that a solution has to meet. Often, a complete and extensive comparison is not needed. The comparator offers the possibility to select only a subset of properties, thus enabling users to regulate the relevance of different properties with respect to their specific needs. By mapping the content structures in XCDL as well as the results from the *Comparator* to the requirements, performance comparisons across different preservation strategies can be defined and recommendations for a solution can be given in an automated way.

The XCL languages presented in this section provide a comprehensive abstract model to describe and express properties of digital objects. The definition of an XCEL allows to describe these properties with a unique vocabulary. The implementation of an XCEL processor enables the extraction of object properties and their representation as XCDL documents. Digital objects described in XCDL can then be processed by interpretation software that compares objects in different representations.

2.6.3 Summary and Outlook

A range of tools exist today for migrating between different file formats. These tools have very specific strengths and weaknesses. Some fail to preserve document layout properly, while others would lose content embedded in objects or fail to preserve structural relations. The evaluation of authenticity of transformed content is a complex task; so is the overall evaluation of suitability of these tools in a particular situation. Digital preservation is thus in need for automated characterisation services that support preservation planning in evaluating potential strategies. These services need an abstract means of describing a document's content in an interoperable, format-independent way.

When comparing the content of two files stored in two different formats, we have to distinguish between the abstract content and the way in which it is wrapped technically. On a very abstract level, this will for a long time be impossible: Whether an image of a hand-written note contains the same 'information' as a transcription of that note in UTF-8 is philosophically interesting, but scarcely decidable on an engineering level. In a more restricted way, a solution is possible if we express the content stored in different file formats in terms of an abstract model of that type of content.

The extensible characterisation extraction and definition languages presented in this section are an important step towards this goal. The extraction language XCEL allows the extractor component to extract the content of any object provided in a format for which an XCEL specification exists. The content is described in the description language XCDL and can thus be compared to other objects in a consistent way. This differentiates the XCL approach from the approach used by JHOVE and similar projects. The XCL does not attempt to extract a set of characteristics from a file, but it pro-

poses to express the complete informational content of a file in a format independent model.

The DFDL language, on the other hand, concentrates on exact typing of data formats for interoperability on the grid. Consider a binary file with the content '00000000 00100000'. Using DFDL, it is possible to express that the file contains an unsigned 16 bit number in big endian encoding, i.e. 32. Parsing tools are being developed to map physical data formats to DFDL [TSSM06]. XCEL is able to express that the file contains a 16 bit number in big endian encoding meaning *imageWidth=32*. The content is then extracted by the extractor using the XCEL specification to produce a representation in XCDL. Thus XCL also intends to describe the semantics of a file.

2.7 Summary and Conclusions

This chapter presented related work in several areas. We outlined the main aspects of digital preservation, giving an overview of the OAIS model and the dominant types of preservation actions, i.e. migration and emulation. We discussed the question of trust in digital repositories and the evaluation of alternative preservaton action components as the central focus of preservation planning. An early case study led to observations about some of the challenges in component selection for preservation planning. We related these peculiarities to a general overview of component selection approaches in a range of scenarios. The last section presented systematic characterisation approaches as a key enabler of automated evaluation.

The next chapter will build on these aspects and present a systematic framework for defining and monitoring preservation plans.

Chapter 3

A systematic approach to preservation planning

3.1 Introduction

The two primary issues in preservation planning, as outlined in the previous section, are the *definition of preservation plans*, including the selection of the most suitable action component as the centerpiece, and *continuous monitoring* of defined plans, the environment, and the repository.

Previous work has led to evidence indicating that utility analysis is a feasible and practical approach to modelling the goals and requirements in digital preservation, and has provided directions for improvement. We have also outlined several related approaches to component evaluation and selection. Taking this into account, we seek a framework and system to enable decision makers select the preservation action component (or combination of components) that does not violate non-negotiable constraints posed by the environment or the organisation, such as legal, budgetary, and technical 'absolute limits', and which of all the available alternatives achieves the 'optimal' score with respect to multiple, potentially conflicting and initially ill-defined preservation goals. This framework shall concretise vague requirements and rely on objective and repeatable measurements to ensure reproducibility. It shall further define action plans for deploying and using the selected component, and provide guidance on how to monitor the operational plans to ensure that any deviation from expected outcomes is noticed and can be reacted upon properly. For all decisions taken, we need full evidence of reasons and documentation to ensure auditable procedures that support trustworthiness.

This chapter will present such a decision making framework and discuss how it supports trust in digital repositories. Section 3.2 defines the scope, content, and structure of a *preservation plan*. Section 3.3 outlines a general framework for evaluation and selection of components in well-defined envi-

ronments, consisting of 5 basic building blocks – requirements, evaluation, analysis, integration, and monitoring. Section 3.4 describes how the general framework is implemented in preservation planning to produce a preservation plan corresponding to the defined structure. Section 3.6 discusses how the approach supports criteria for trustworthy repositories. The subsequent chapters will then describe tool support and experiences in applying the framework in a series of case studies. Chapter 4 presents the tool architecture we have developed. Chapter 5 discusses examples of applying the method and tool in practice, while Chapter 6 demonstrates the collection of evaluation data in a controlled environment through automated measurements and discusses the coverage of measurements in the light of real-world case studies.

3.2 What is a preservation plan?

3.2.1 A pragmatic definition

An important distinction has to be made between concrete preservation *plans* and high-level *policies* which are generally made at an institutional level and regulate fundamental constraints and strategies.

There is a number of documents available which lay out policies for digital preservation. The Erpanet policy tool supports policy definition on an institutional level [Erp03]. A recently published JISC funded study on digital preservation policies outlines a model for digital preservation policies with the aim of helping institutions develop appropriate digital preservation policies [Nei08].

The 'ICPSR Digital Preservation Policy Framework'[1] defines high-level factors and makes the institution's commitment explicit. The British Library's Digital Object Management team has defined a preservation plan for the Microsoft Live Book data, laying out the preservation policies for digitised books.[2] The policy defines high-level responsibilities and certain formats which are subject to continuous monitoring, but does not specify actionable steps. The self-assessment tool developed at the Northeast Document Conservation Center[3] aids in preservation planning, however at a similarly high conceptual level.

These documents define abstract, high-level policy concerns. While they provide very useful and important guidance, they are more setting a framework for concrete planning than actually providing actionable steps for ensuring long-term access. Examples of policy elements that are covered include

[1] http://www.icpsr.umich.edu/DP/policies/dpp-framework.html

[2] http://www.bl.uk/aboutus/stratpolprog/ccare/introduction/digital/digpresmicro.pdf

[3] http://www.nedcc.org/resources/digital/downloads/DigitalPreservationSelfAssessmentfinal.pdf

'Preservation action must be open source' and 'Cost of preservation action must not exceed estimated value of digital object'.

A preservation plan, on the other hand, is seen on a more specific and concrete level as specifying an *action plan* for preserving a specific set of objects for a given purpose. For reasons of traceability and accountability, this also needs to include the reasons underlying the decisions taken. We thus rely on the following definition, which has been adopted by the Planets project[HPB+08].

> A preservation plan defines a series of preservation actions to be taken by a responsible institution due to an identified risk for a given set of digital objects or records (called collection). The Preservation Plan takes into account the preservation policies, legal obligations, organisational and technical constraints, user requirements and preservation goals and describes the preservation context, the evaluated preservation strategies and the resulting decision for one strategy, including the reasoning for the decision. It also specifies a series of steps or actions (called *preservation action plan*) along with responsibilities and rules and conditions for execution on the collection. Provided that the actions and their deployment as well as the technical environment allow it, this action plan is an executable workflow definition.

3.2.2 Elements of a preservation plan

A preservation plan thus should contain the following elements:

- Identification,
- Status and Triggers,
- Description of the institutional setting,
- Description of the collection,
- Requirements for preservation,
- Evidence of decision for a preservation strategy,
- Costs,
- Roles and responsibilities, and
- Preservation Action Plan.

We will discuss these elements in detail in the following sections, which are largely taken from our definition of a preservation plan as presented in [BKG+09].

Identification

A preservation plan should be uniquely identified so that it can easily be referred to and retrieved.

Status and Triggers

The status of a plan includes both the planning progress – whether a plan is currently being defined, awaiting approval, or has already been deployed and is active – and the triggers which have led to the definition or refinement of the plan.

Specifically, the following events may trigger a planning activity and should thus be included in the documentation of the plan.

- *New Collection.* This is the most common event, where a preservation plan is created from scratch for a new collection for which no plan was previously defined.
- *Periodic Review.* Periodic reviews of existing preservation plans are needed to verify the appropriateness of plans, and to improve and further develop existing plans. A periodic review, e.g. every 3 to 5 years, should re-iterate the planning activity, taking into account new developed preservation strategies, and seek to verify and potentially improve established plans.
- *Changed Collection Profile, Environment, or Objective.* Complementary to the documentation of recorded events that triggered an activity, the completed preservation plan thus also contains a specific definition of events that should trigger a revision of the preservation plan. This enacts a monitoring of those aspects of the environment that are considered to be of particular relevance or particularly prone to change. These triggers hence apply to revised plans where an alert has been raised by a monitoring function, indicating that a plan needed to be updated to reflect changed conditions. Section 3.5 discusses them in detail.

This section of the preservation plan further contains several key dates and relations to other plans, which normally are referring to the events discussed above.

- *Valid from* defines the date on which the plan becomes active.
- *Based on* identifies a preservation plan on which the plan is based. This could for example be a plan that was overridden because of a changed objective.

- *Replaced by*, *Replaced on date* and *Invalidated on date* are the corresponding counterparts which create a bi-directional reference between related preservation plans.
- *Approved by* and *Approved on* document the responsible approval of the plan.

Description of the institutional setting

This part documents the reference frame of the preservation plan, the main context in which the planning activity takes place and in which the plan needs to be functional. Thus it needs to cover a broad range of high-level influence factors that have an impact on the decisions taken in defining the plan. Prime examples of aspects that are considered essential in this context include

- the *mandate* of the repository, e.g. the mission statement of the organisation;
- a description of the *designated community* for the considered collection; and
- references to applying legal, operational, and preservation policies.

Further of interest are for example

- a description of relevant organisational procedures and workflows;
- references to contracts and agreements specifying preservation rights; or
- references to agreements of maintenance and access.

For a well-founded thorough description of the institutional setting, a clear understanding of the institution's designated user community and policies is necessary, as both are important parameters for decisions throughout the preservation planning process. A detailed usage model which describes how users work with their collection and which priorities they have supports the specification of requirements and brings to light the users' priorities. Policies describe how the institution is carrying out its mandate and define organisational characteristics and goals of the repository. Particular policies may also constrain the range of potential preservation actions to be considered.

Description of the collection

The collection is the set of digital objects or records for which a preservation plan is created. It can be technically homogeneous (e.g. one file format), but might also consist of different types of objects or file formats. It can also be based on a *genre* in the sense of "all emails in my repository". Technically speaking, it refers to all objects that shall be treated with the same tool with identical parameter settings during the application of preservation actions.

This section includes

- an identification of the objects that shall be preserved, such as persistent identifiers that can be resolved in a repository or a unique name identifying the set of objects;
- a description of the *type of objects* mentioning general characteristics such as the contained class of objects and the file format(s); and
- *sample objects* that are representative for the collection and thus can be used for the evaluation process. This should include the actual objects and a description of their well-understood properties as well as their original technical environment.

Requirements for preservation

This section shall describe as detailed as possible the requirements that are underlying all preservation planning decisions.

Relevant requirements include a specification of the significant properties of the objects under consideration, to ensure that the potential effects of applying preservation actions are evaluated against the clearly specified aspects of objects and potential impacts are considered during the decision process.

They will usually also cover aspects such as desired process characteristics, cost limits that need to be taken into account, or technical constraints that have to be considered. Potential requirements and a specific approach of defining these in a hierarchical form are discussed in detail in Section 3.4.3.

Evidence of decision for preservation strategy

Evidence plays an essential role in establishing trust in digital repositories; evidence-based decisions and proper documentation foster transparency and support the building of trust[RM06, TO07]. This section is thus considered vital to guarantee and document that an accountable decision has been made.

The following elements are considered necessary to establish a chain of evidence that enables accountability and the tracing of decisions to link them to influence factors and assess the impact of changes further on.

- A *list of alternative actions* that have been closely considered for preservation. This should include the selection criteria that were used for narrowing the list of alternatives down from the total set of available approaches to a 'shortlist'.
- *Evaluation results* that take into account how the considered alternatives fulfil the specified requirements and document the degree of fulfillment as objectively as possible.
- A documented *decision* on what preservation strategy will be used, including the reasons underlying this decision.
- A documentation of the *effect* of applying this specific action on the collection, explicitly describing potential information loss.

Costs

This section specifies the estimated costs arising from the application of this preservation plan. A quantitative assessment relying on an accepted cost model such as LIFE2[ADM+08] is desirable.

Roles and responsibilities

This section specifies the responsible persons and roles carrying out, monitoring and potentially re-evaluating the plan.

Preservation Action Plan

The preservation action plan specifies the concrete actions to be undertaken in order to keep the collection of digital objects alive and accessible over time.

A preservation action might be just the application of a single tool to a set of objects, but can also be a composite workflow consisting of multiple characterisation and action services. In this sense, the preservation action plan specifies two main aspects: the *When* and the *What*.

- Triggers and conditions specify when the plan shall be executed, as well as specific hardware and software requirements and other dependencies.
- The *Executable Preservation Plan* specifies the actions that will be applied to the digital objects and should also include automated mechanisms for validating the results of the actions, i.e. automated quality assurance, wherever possible. The concrete elements of this part depend on the system architecture of the target environment where it shall be deployed. It can for example be an executable web service workflow deployable in the Planets environment [KSJ+09].

- Other actions needed might include reporting and documentation of the steps performed.

3.2.3 Summary

This section described the main components of a preservation plan. The structure of the plan comprises not only the action steps to be taken and corresponding responsibilities, but also documents the reasoning behind the decisions and shall include the complete evidence base of decision making.

The next two sections describe a systematic method of defining preservation plans that conform to this structure through a repeatable and transparent workflow that supports the automated documentation of decisions.

This method will be introduced in two stages: We first describe a framework for component evaluation and selection based on controlled experimentation and automated measurements. This framework has been developed in the context of digital preservation, but is applicable to a wider range of well-defined domains, as will be discussed. Section 3.4 will then describe the domain-specific instantiation of the framework in detail and explain how it is used to create and monitor preservation plans.

3.3 A framework for automated component evaluation and selection

3.3.1 Introduction

As discussed in Section 2.5, most component evaluation and selection approaches are geared towards flexibility and applicability in a wide range of domains, and assume that component selection is a one-off procedure carried out within a development effort to build a new system. In these scenarios, evaluation of candidate components against requirements can be done in a largely manual way, which usually implies, but also allows for, high levels of complexity.

However, some of the outlined assumptions are beginning to change.

The last decade has seen significant shifts in a number of determining factors for software component evaluation, selection, and integration. Service orientation has become the primary paradigm for decoupling components to build and integrate complex systems; the sheer number of software components in any given system has soared; and the question of trust has become central to the assessment and selection of software, especially of services. Garlan et. al. recently re-emphasise that the main challenges for software reuse today are trust, dynamism, architecture evolution, and architecture lock-in [GAO09]. In complex environments with changing requirements, subjective human judgement of software quality and the reliance on declared capabilities of components cannot be considered sufficient evidence. They

cannot replace objective measurements obtained in a controlled environment as the basis of decision making. As Terzis recently stated,

> ...the modern view of trust is that trustworthiness is a measureable property that different entities have in various degrees. Trust management is about managing the risks of interactions between entities. Trust is determined on the basis of evidence ... and is situational – that is, an entity's trustworthiness differs depending on the context of the interaction [Ter09].

If an entity's trustworthiness has to be validated in the context of an interaction, we need to do so in a controlled environment where the varying parameters are known and the outcomes repeatable, reproducible, and measurable.

In many cases, component evaluation is not done once when constructing a system, but needs to be managed as a recurring operation where components need to be monitored to detect mismatches and eventually reconfigured or replaced when they prove to be unsuitable in a continuously changing environment. This usually occurs in environments where the functionality offered by competing components or services is sufficiently standardised and well-defined, so that non-functional quality attributes are of primary importance. Examples are data transformation routines in large-scale information systems such as content migration in digital library systems for content preservation [HC06, FBR07, SBNR07, BFK+08], or machine translation modules [Joh09].

We have seen that in digital preservation, the functionality of preservation action components is very focused and well-defined. Furthermore, the level of responsibility of institutions such as national archives implies that a trusted evaluation and selection process is vital and needs to be auditable. This is emphasised by auditing initiatives for trusted repositories such as TRAC [TO07]. Similar requirements can be found in numerous application domains such as compression tools or sort and search in high-dimensional index structures. For example, in machine translation, components have similarly focused and well-defined functionality, and techniques for automated evaluation of translation quality are being developed [PRWZ02, Dod02, LRL05]. Again, thorough evaluation and continuous monitoring of a translation component is required to cope with e.g. topic drift in the source documents.

The concepts presented in this chapter and the system described in Chapter 4 support auditable selection and continuous monitoring of components via general and domain-specific measurements. They can be applied beneficially in settings sharing the following characteristics:

1. *Homogeneous functionality.* – The functionality of components is homogeneous and well-defined. Competing tools will provide the same

functionality, allowing the creation of dedicated evaluation modules for these components.

2. *Continuous evaluation and monitoring.* – The selection process has to be repeated regularly, potentially leading to a reconfiguration or replacement of components.

3. *Transparent and auditable decisions* are necessary to support the critical requirement of trust in software components and services. Specific quality requirements might be negotiable, but a thorough and objective documentation about the information that was available at the time of decision making is of vital importance. Thus, decision making and component selection procedures need to be fully transparent and reproducible to provide sufficient levels of accountability.

These peculiarities on the one hand have the effect that the existing approaches for evaluation and selection do not fully satisfy the needs of the scenario. On the other hand, it provides opportunities for leveraging the scale of the problem space to employ automated evaluation techniques for the selection process. We argue that it is possible to apply automated measurements of quality attributes to a majority of the criteria by conducting controlled experiments with the candidate components, provided that the functionality offered by these components is homogeneous.

The rest of this chapter describes the decision framework we use for component evaluation and selection. It is based on utility analysis and controlled experimentation and supported by a distributed architecture of registries and services to enable the controlled and automated evaluation of software components. We will outline the high-level workflow and the main concepts behind the method, and describe in detail how the framework is being put to use in digital preservation.

3.3.2 Workflow

Figure 3.1 abstracts the principal steps and building blocks of the evaluation environment. The steps of the workflow are similar to the general COTS selection process [MRE07] (GCS), but include a final stage where the software product is integrated into the system, and a continuous monitoring activity after product integration:

1. Define requirements,

2. Evaluate components,

3. Analyse results,

4. Integrate product, and

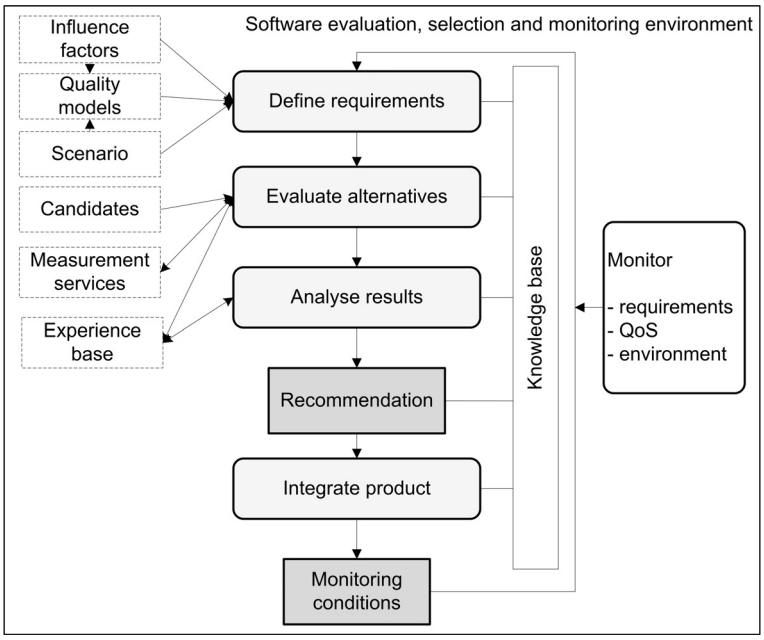


Figure 3.1: Software evaluation, selection and monitoring

5. Monitor requirements, quality of service, and the environment.

This corresponds to standard component selection workflows. The main differences are that importance weightings are refined in the analysis phase, after measurements have been taken, and that monitoring and re-evaluation are an integral part of the method, relying on automated measurements and QoS specifications. The next sections will give a high-level overview of the main stages of the workflow.

3.3.3 Requirements definition

Requirements definition relies on a multi-objective decision analysis method based on utility theory [vNM44], which has been applied to a wide range of selection problems [KR93, RFC94, WSA+01, RR04, JMRIR07]. Goals and criteria are specified in a hierarchical manner, starting at high-level goals and breaking them down until quantifiable criteria are found at the bottom level of the hierarchy. The problem of incommensurable values [Hsi09] is tackled by defining a *utility function* which transforms measured values to a common utility value which can be compared across alternatives and aggregated in the goal hierarchy. Relative importance factors on each level of the hierarchy model the preferences among the stakeholders.

Section 3.4.3 will discuss the requirements approach in detail.

3.3.4 Evaluation and analysis

Requirements evaluation takes advantage of the homogeneity of the problem space and follows an empirical approach. Some of the evaluation criteria may be retrieved from databases holding confirmed information about static attributes such as the price or the licensing model of candidates. Specific quality criteria such as *accuracy* of operations or average *processing speed* for certain input data need to be evaluated in an evidence-based, empirical manner. Candidate tools are executed in a controlled environment, providing a thorough evaluation and evidence base in realistic experiments.

In principle, three categories of criteria need to be evaluated for each component:

1. Statically defined criteria can be retrieved from a trusted database holding product information not subject to change.

2. Process- and performance-related characteristics can be measured automatically during tool execution in the experiments stage. The tools are invoked through a monitoring framework which is able to measure general process-related characteristics such as performance [BKK+09a].

3. Criteria specific to the application domain, such as accuracy, are measured by an extensible architecture of measurement plugins, which is described in Chapter 6.

The evaluation of experiment results leads to a requirements tree fully populated with evaluation values in the respective scale of each criterion. This is the basis for the third phase, which corresponds to the GCS step *Analyse data and select product.*

Analysis of results consists of three steps:

1. Transform values to a uniform scale,

2. Set importance factors, and

3. Analyse the outcomes to arrive at a candidate recommendation.

In order to apply aggregation and comparison of values over the tree hierarchy, a *utility function* is defined, which maps all evaluation values to a uniform target scale of commonly 0 to 5. Hereby, 5 is the optimum value, whereas 0 denotes unacceptable performance that serves as a drop-out criterion.

Existing component selection methods usually favour the definition of target ranges for requirements before the actual values and capabilities of potential components are known. They rely on negotiation of conflicting values during package selection [CFQ07]. Transforming actually measured values after knowing the results allows trade-off decisions and negotiation of

acceptable values based on a trustable knowledge of reality. In bid evaluation, the actual evaluation values should be anonymised to avoid decisions to be influenced by internal biases.

The result of the transformation step is a fully populated, evaluated and transformed requirements tree with default weighting. The next step is to revise the default weights to reflect the actual priorities and preferences of the stakeholders.

There has been considerable discussion on the question of importance weighting in component selection methods. Several methods using weighted scoring methods have earned criticism for the fact that weight settings need to be specified upfront, in a situation where little or nothing is known about the actual performance and differences of candidate components. Furthermore, the reduction to a single number and the corresponding ranking is considered too simplistic by many. The Analytic Hierarchy Process on the other hand is often considered too effort-intensive and complex, since the number of pairwise comparisons is exploding with the size of the requirements tree [ND02, PRS09].

In the presented approach, relative importance factors are balanced on each level of the tree after evaluation values for all candidates are known and utility functions have been defined. This deviates from standard utility analysis workflows, but has proven more useful in the considered selection scenario in numerous case studies. In order to safeguard against potentially negative effects of minor variations in the weighting on the stability of decisions, a sensitivity analysis is performed. In this step, several hundred iterations are automatically computed, randomly varying weights in a certain percentage margin around the given value to identify potential changes in the ranking of components. This results in rank robustness measures on all levels of the goals hierarchy that can lead to a closer analysis of critical aspects that are sensitive to variations. We are currently investigating the narrowly foccused use of robust, but effort-intensive ranking models such as AHP for ranking a small number of critical high-level factors as well as competing factors that show high sensitivity during the analysis.

In the final step, visual analysis of results allows a comparison of performance values not only on the root level, but on all levels of the tree hierarchy. The complete evaluation, transformation and aggregation is used as an evidence base to support the decision for recommending one of the candidate components. The method furthermore allows the selection of multiple components that are considered to be complementary, should none of the alternatives alone completely satisfy the needs to a sufficient degree.

3.3.5 Integration and monitoring

Integration is the final stage of the workflow, where the component's interface to the system under consideration is defined. This includes explicit state-

ments about evaluation conditions that need to be monitored continuously in the operation of the chosen software. These conditions can be largely deducted from the requirements and defined in service level agreements [KL03].

The problem of *architectural mismatches* is often encountered in CBSD, primarily when dealing with coarse-grained components where controlled experimentation is rarely applicable. Selecting a seemingly optimal component based on an analysis that focuses on functional issues can lead to serious cost- and budget overruns when implicit architectural assumptions prove to be incompatible and conflicting [GAO95].

Our approach does not lead to the recommendation of a component without fully testing it in a controlled environment, so this aspect is less likely to have an effect; mismatches are prevented or detected early [GAO09]. If a component can be successfully evaluated on real data in a controlled environment under realistic conditions, the risk of a serious mismatch on either an architectural or a lower technical level is quite limited. Furthermore, platform constraints are part of the evaluation procedure. If a component cannot be evaluated – for example due to incompatibility with a server environment or protocol – this is a documented outcome of an experiment. In the absence of a re-run experiment, it leads to rejection of the component. The method and tool allow feedback loops, should experimentation lead to the discovery of such problems. Evaluators then return to earlier stages to refine requirements, reconsider acceptable values, or re-balance importance factors. For example, if it turns out that the optimally performing tool is too slow, but all other components produce inacceptable results quality-wise, it may be necessary to reconsider performance requirements or reduce expectations in terms of quality.

3.3.6 Discussion

Component selection and evaluation is a continuous problem space ranging from CBSD to web services and other dynamic scenarios as summarised in Table 2.1 in Section 2.5. Figure 3.2 contrasts in a simplified perspective three scenarios for component selection: Web service selection on the dynamic end of the spectrum is highly dynamic, deals with fine-grained and precisely specified components, and relies entirely on automated evaluation. CBSD on the other end deals with any granularity of functions and is less dynamic. The degree of trust needed and the importance of NFRs certainly vary in each case, but are in general not as high as for example in digital preservation. In our method, the selection and integration is followed by continuous monitoring. Deviations from specified expectations lead to a re-evaluation that can result in a reconfiguration or replacement of the component. The method is applicable to fine-grained components with a homogeneous, well-defined feature set, and the focus lies on automated measurements obtained in a controlled environment.

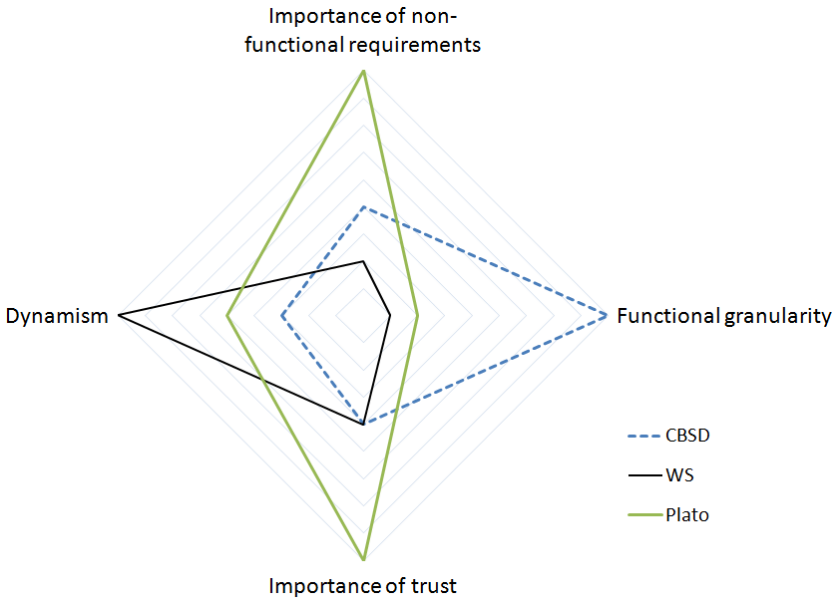


Figure 3.2: Comparison of foci in component selection scenarios

The next section will illustrate in detail how this framework supports the concrete evaluation procedures and measurements in digital preservation. We will translate the high-level workflow into concrete actionable steps and discuss examples.

3.4 The preservation planning workflow

We have discussed which aspects should be covered by preservation plans as opposed to general policies, and described the desirable components of a preservation plan. What is clearly needed is a method of specifying, monitoring and updating these preservation plans in a transparent, accountable and well-documented way.

This section proposes such a method. It is based on earlier work described in Section 2.4.2 which has been revised and extended, and implements the general component selection approached described in Section 3.3 for digital preservation planning. This section describes the primary planning workflow for evaluating potential actions and specifying concrete preservation plans.

Two key issues have to be addressed by a preservation planning workflow: Evaluating potential actions and specifying concrete steps to be taken. Evaluation and selection of the most suitable component is a specific instantiation of the component selection framework described in Section 3.3. Based on the product selection, a concrete plan is defined which corresponds

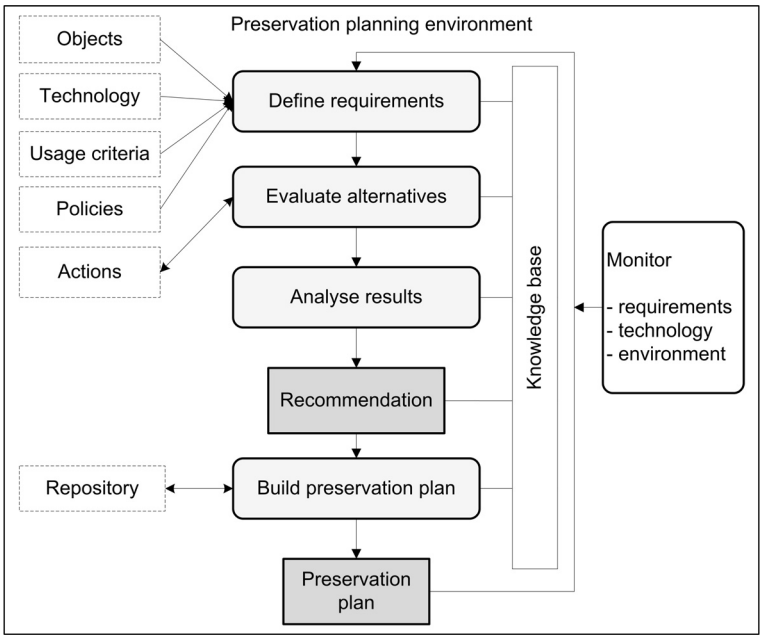


Figure 3.3: Preservation planning environment

to the definition discussed in the previous section.

The resulting workflow thus consists of five phases:

1. Define requirements,
2. Evaluate alternatives,
3. Analyse results,
4. Build preservation plan, and
5. Monitoring.

Figure 3.3 illustrates the preservation planning environment, putting the high-level workflow in the context of the main environment factors to which it relates. The four primary phases result in a working preservation plan that can be continually executed. Building the preservation plan corresponds to the generic step *Integrate product* in the method described in Section 3.3. An ongoing monitor function is necessary to ensure the ability to adapt to detected changes in either the environment, the technologies used in operations, or changing objectives. This results in a continuous circle of revisions to preservation plans and enables the repository to react accordingly to the inevitable changes to be expected.

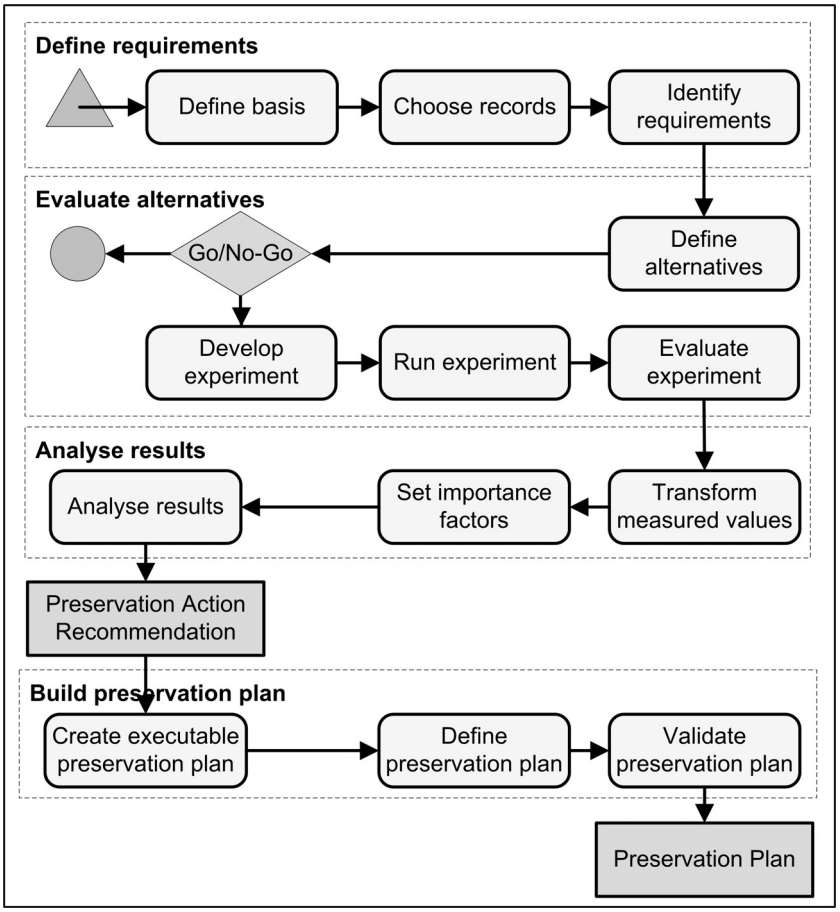


Figure 3.4: Workflow for creating a preservation plan

Figure 3.4 shows the concrete steps within the high-level workflow for creating a plan, which the next sections will discuss in detail. Section 3.5 will focus on the issue of monitoring.

3.4.1 Define requirements: Define Basis

The first phase of the workflow lays out the cornerstones of the planning endeavour. It starts with collecting and documenting the influence factors and constraints on possible actions and procedures, then describes the set of objects under consideration, and finally defines the complete set of requirements to be taken into account. The first step thus documents the main elements underlying the planning activity. It collects and documents the primary influence factors constraining the decision space, and thus lays the foundation for a thorough documentation and makes sure that all relevant aspects are established and considered. This covers the headings 1 to 4 of

Category	Example policy elements
Formats	All formats must be ISO standardised
	Formats must have open specification
	Image formats must support lossless compression
	Formats shall not support encryption
	Formats must be supported by current tools
Signficant properties	Sacrifice usability for authenticity
	Sacrifice structure for software independency
	Access copy shall be archival copy
User access	Documents shall (not) be editable
	Text shall be searchable
Infrastructure	Action components must work in host environments
	Other compatibility constraints
Strategies	Actions must be migration
	Actions must be emulation
	Only use lossless compression
Process	Accept irreversible migration
	May delete original
	Maximum costs per object
	Costs must not exceed estimated value per object

Table 3.1: Example policy elements

the preservation plan as described in Section 3.2.

Case studies have revealed that a comprehensive definition of influence factors is an important prerequisite for successful planning. The documentation of constraints that might limit the choice of applicable options in this stage simplifies and streamlines the selection procedure and ensures that the outcome is indeed in line with the needs of the institution.

In this step, the preservation planner documents applying institutional policies, legal regulations, and usage criteria that might affect planning decisions for preservation. This may happen in an unstructured form, but preferably these factors are captured in a more formal way, making it easier to derive decisions in the respective workflow steps. Examples include policies that define permitted file formats for (re-)ingest, and policies related to intellectual property rights and legal access regulations. Further important policy elements pertain to characteristics of the preservation action, whether preservation actions that are open source shall be preferred, or if just a specific class of preservation action may be applied, such as emulation. (The latter can occur in cases where the institution does not have the copyright and thus any modifications of the digital objects are prohibited.)

Tabel 3.1 shows some fundamental criteria that should be documented. Most of these describe constraints that will have to be considered in selecting

the most suitable action. Examples include constraints on file formats – for example, formats may have to be standardised by ISO or a different recognised body, or must not support encryption. In many organisations, homogenising the formats is a high priority; this may imply that only certain formats are considered for each type of object. On the other hand, there might be known preferences about access modes and desired features of user access, such as searching and editing for further use, or a strong preference to disable editability for reasons of authenticity and fixity. These preferences vary corresponding to the organisational context and the designated user community's preferences. On a technical level, compatibility constraints may be known; and on a process level, there may be known limits about irreversibility and maximum costs.

Furthermore, the event that led to the planning procedure is documented. As described in Section 3.2, planning can be triggered by a new object type that is accepted, or a change in collection profiles, objectives, or the environment.

3.4.2 Define requirements: Choose records

The second step describes the set of objects that form the scope of the current plan, and selects a subset of representative objects for experimentation, as required in Section 4 of the preservation plan.

A general description of the characteristics of the set of objects, called *collection*, includes basic properties such as the size of the collection, the class of objects, and the object formats they are currently represented in. While this can be done in a manual descriptive way, a formal representation is desirable. Collection profiling tools can provide automated descriptions of the technical characteristics of objects. An example of such a profiling service is described in [BCH+07].

Characteristics of interest include object formats, file sizes and their variation within the collection, but also aspects such as an assessment of the risks of each object type and each object, thus leading to a risk profile of the collection. Sometimes it is also necessary to document an estimated collection value and time horizons for preservation.

Since a complete evaluation of the quality of preservation action tools is infeasible on the potentially very large collection of objects, the planner selects representative sample objects that should cover the range of essential characteristics present in the collection at hand. To reduce effort to a minimum, this subset should be as small as possible. However, the samples are used as a representative set for testing the effects of applying preservation actions to the whole set of objects. A complete and thorough evaluation of the quality of preservation actions relies heavily on the completeness of features present within the test set. Hence, the set must be as large as necessary to cover the variety of essential characteristics on the technical level.

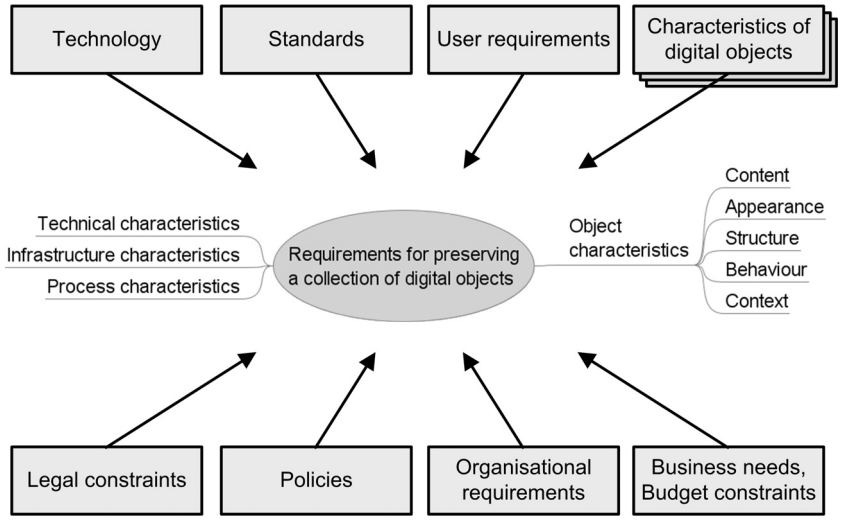


Figure 3.5: Influence factors

Depending on the degree of variance within the collection, typically between 3 and 10 sample objects are selected. For these samples, an in-depth characterisation is performed, describing the significant properties and their technical characteristics such as their name and provenance, the file format, and specific risk factors.

3.4.3 Define requirements: Identify requirements

Requirements definition is the heart of preservation planning. It is the basis for the decisions to be taken and documents the priorities and preferences of the institution. This step enlists all requirements that the optimal digital preservation solution needs to fulfil, as in Heading 5 of the preservation plan (cf. Section 3.2.2).

We rely on a variation of multi-objective decision analysis based on utility theory. Goals and criteria are specified in a hierarchical manner, starting at high-level goals and breaking them down until quantifiable criteria are found at the bottom level of the hierarchy. The problem of incommensurable values [Hsi09] is tackled by defining a *utility function* which transforms measured values to a common utility value which can be compared across alternatives and aggregated in the goal hierarchy. Relative importance factors on each level of the hierarchy model the preference structure of the decision makers.

Requirements are collected from the wide range of stakeholders and influence factors that have to be considered for a given institutional setting. This may include the involvement of curators and domain experts as well as IT administrators and consumers. The requirements are specified in a quantifiable

way, starting at high-level objectives and breaking them down into measurable criteria, thus creating an *objective tree* (also referred to as *requirements tree)* which forms the basis of the evaluation of alternative strategies.

Figure 3.5 shows the root levels of such a tree, together with the factors that are influencing the requirements definition. Some of these high-level factors have been documented in the first two steps; in this step, they are informing the concrete specification of the objective tree.

Requirements definition has proven to be the most critical and complicated stage of the planning procedure. An incomplete requirement specification leads to a skewed evaluation and potentially wrong decisions. On the other hand, curators tend to exhibit a reluctancy to quantify their preferences, and especially try to avoid questions such as *What is the loss I am willing to accept?* which are of central importance.

The complexity involved in specifying goals and breaking them down to concrete, quantifiable criteria is a considerable challenge. However, through iterative refinement of abstract goals such as *I want to preserve these objects exactly as they are* towards more concrete requirements (*The size needs to remain unchanged*) we ultimately arrive at measurable criteria such as *The image width, measured in pixel, needs to remain unchanged.* We subdivide objectives into lower-level objectives, which can be seen as means to an end, clarifying the general objective and specifying it more precisely [MH67]. This is combined with bottom-up criteria collection. The ideal is to arrive at a set of attributes that are complete, operational, decomposable, nonredundant, and minimal [KR93].

This procedure benefits from a broad involvement of stakeholders to elicit all necessary pieces of information, correctly document institutional policies and priorities, and establish constraints. A common approach is, in the spirit of Socratic discovery, to elicit the requirements in a workshop setting where as many stakeholders as feasible are involved, moderated by an experienced preservation expert. This involvement has to avoid skewed decision priorities incurred by dominant stakeholders and needs to be managed carefully in the beginning by an expert responsible for modelling the requirements in the objective tree. As an organisation is successively repeating the planning procedure for different types of objects, it is gaining expertise and experience and accumulating known constraints. These are documented in its knowledge base, and the need for constant stakeholder involvement and moderation gradually declines.

It is, of course, also possible to perform the elicitation of requirements in a sequential order, having all individual stakeholders list their specific requirements individually, and then integrate them in to a single objective tree. However, different aspects raised by some stake holders in a discussion process often lead to a better understanding of the various characteristics of the objects as well as the preservation process and the forms of usage. Note, that - contrary to conventional requirements elicitation, where trade-

offs between requirements are defined in such a workshop setting - this is not the case here, as the focus is on complementary requirements and views on the preservation process and the objectives it shall meet. Trade-offs and weightings are performed in the third stage of the process (cf. Section 3.4.9).

On a practical level, two tools have been very useful for the requirements elicitation process: sticky notes and mind-mapping software. Sticky notes and flip charts as traditional tools for brainstorming activities have the benefits of allowing everyone to act at the same time. Mind maps provide the better overview of the current state of requirements for all participants and allow a moderator to channel the discussion process. Often, a combination of both tools is the most productive approach. Using these tools, the requirements are structured in a hierarchical way, starting from general objectives and refining them via more specific requirements to arrive at measurable criteria that a successful digital preservation solution has to meet. This structure is further referred to as requirements tree or "objective tree", i.e. a tree capturing the objectives to be met.

Existing quality models provide sophisticated specification of quality criteria, metrics, and their relationships [ISO01, FC03, CFQ07]. While these concepts provide for powerful modelling tools, they are difficult to use, especially for decision makers not familiar with them. In the component selection tool DesCOTS, the task of defining these models is thus transferred to COTS evaluation experts that model domains and evaluate products [GCFQ04, QFLP06, QFLP05]. In contrast, the hierarchical definition of requirements in the approach described here is not as formally strict as these models; hence, the requirements can be specified by domain experts themselves, without dedicated external assistance. Depending on the level of experience with the selection scenario in a specific organisation, the requirements definition process ranges from a simple reuse and customisation of existing quality models and objective trees to interactive group sessions, where software support aids in the definition process.

On the bottom level of the requirements tree, measurable criteria have to be defined such as *processing speed per megabyte* measured in milliseconds, or *output format is ISO-standardised*. These criteria are annotated with information on how to obtain the actual measurement data for the candidates during the evaluation stage. Several scales sometimes used in multi-criteria decision making approaches are not employed. For example, 'forced ranking', i.e. direct preference measurement as establishing a preference order for the considered options, is not used, since the level of reasoning and provided documentation is not considered sufficient. Traceability and transparency of evaluations call for breaking down such a ranking into the decisive aspects of which it is composed, and evaluating each aspect separately.

While the resulting objective trees usually differ through changing preservation settings, some general principles can be observed. At the top level, the objectives are often organised in four main categories – characteristics

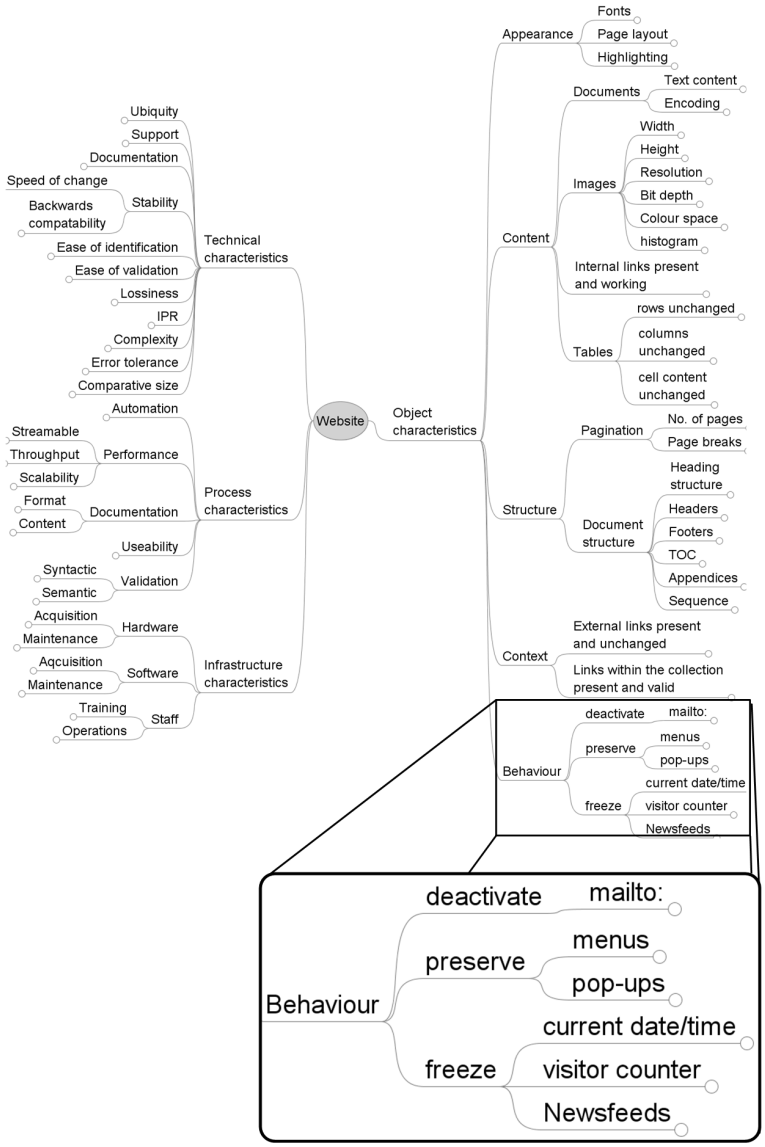


Figure 3.6: Requirements specified in an objective tree

of the objects, the records, and the process, and requirements on costs.

- *Object characteristics* describe the visual and contextual experience a user has when dealing with a digital object. These characteristics are often referred to as *significant properties*. A common way of describing them is to consider the five aspects “Content", “Context", “Structure", “Appearance", and “Behaviour" [RB99].

  Figure 3.6 highlights an example of specifying desirable *transformation*

of behaviour when preserving a web archive of a national domain crawl. The tree contains the requirements for preserving a collection of static web pages containing documents and images. The branch *Behaviour* is divided into three different groups of criteria: *deactivate, preserve*, and *freeze*. This reflects the preferences of the archive that some functionality, such as menu navigation, is needed for properly accessing the web pages, while most active content shall be disabled or frozen. For example, visitor counters shall be preserved in the state they had at the moment of ingest, rather than preserving their activity and to continue counting within the archive. (This scenario may well be of interest for a different designated community of e.g. internet historians who want to analyse the technical principles of how counters were implemented in earlier days.)

This category is also an exemplary case of conflicting requirements, which often occur. For example, a purpose-built emulator might deliver a perfect representation of significant properties including behaviour, but at high costs, not scalable, or not accessible on the web. On the other hand, migration on demand saves storage space, but requires a certain infrastructure to deliver fast access to thousands of clients. Decision makers may have to decide between a very slow tool performing accurate transformation including integrity checks, but scaling poorly, and a highly scalable and reliable tool that loses certain properties in transformation in a predictable way and produces standardised progress reports that can directly be integrated into the repository workflow. It is hardly possible to fulfil all criteria; trade-off decisions are common. The explicit hierarchical criteria specification makes these decisions transparent and clear, and supports the direct comparison of strengths and weaknesses.

Recently, several projects such as INSPECT[4] have presented detailed analyses of the significant properties of different categories of objects. InSPECT proposed to refer to “Function”, “Behaviour", and “Structure" [GKM09], following a framework developed to assist engineers and designers in creating and re-engineering systems [Ger90], and analysed significant properties of vector images, moving images, e-Learning objects, and software [HYK08]. These analyses can provide a very valuable input to this aspect of requirements specification. On the other hand, the automated characterisation of the sample objects defined in the previous step further supports the specification of significant technical properties from a bottom-up perspective.

- *Record characteristics* describe the foundations of a digital record, the context, interrelationships and metadata. This may include simple,

[4] `http://www.significantproperties.org.uk/`

but often overlooked, linking requirements, such as the fact that file names need to remain unchanged or consistently renamed across subcollections if they form the basis for cross-referencing or inclusion.

- *Process characteristics* describe the preservation process itself, for example the procedure of migrating objects. These characteristics include the complexity of applying preservation action components or their performance, scalability, and usability, but should equally cover aspects such as documentation or the degree of automated validation.

  The definition of process characteristics is particularly dependent on the specific context in which the preservation process is taking place. The technical environment may effectuate specific requirements on the interoperability of tools, while institutional policies or legal regulations may enforce specific licensing requirements or require a particular degree of automated documentation. Thus the institutional and technical context and constraints posed by it have to be considered carefully.

- *Costs* have a significant influence on the choice of a preservation solution, but are inherently hard to quantify. Ultimately the Total Cost of Ownership (TCO)[5] is the guiding figure for deciding whether or not a preservation strategy meets the needs of an institution within the constraints of its budget. Instead of providing a single numeric criterion which is extremely complex to quantify, costs might also be defined as *infrastructure characteristics*, putting an emphasis on cost factors instead of the resulting figures for cost estimates. These cost factors can then be further broken down to cover hardware, software, and staff costs, as shown in Figure 3.6.

An essential step of requirements definition is the assignment of measurable effects to the criteria at the leaf level of the objective tree. Wherever possible, these effects should be objectively measurable (e.g. € per year, frames per second, page orientation, bits per sample) and thus comparable. Care has to be exercised not to mistake significant properties of objects with the criterion that they shall be left unchanged. Properties such as *image width* measured in pixels will have to be compared, for example for equality, and the resulting criterion will be measured on a Boolean scale.

In some cases, (semi-) subjective scales need to be employed. For example, the quality of documentation that is available for a file format or a tool should not be judged by the number of pages alone; instead, a subjective scale such as *excellent, good, average, poor, very poor* could be used. Similarly, the *openness* of documentation of a file format could be one of *fully standardised*; *openly published*, but not standardised by a recognised

[5] `http://amt.gartner.com/TCO/MoreAboutTCO.htm`

body; and *proprietary*. Along the same lines, the *stability* of a format can be measured in revision time intervals and backwards compatibility.

The assignment of measurable effects to criteria can also align them with characteristics that can be automatically extracted from objects to automate the evaluation procedure. Existing software tools such as JHOVE[6] allow automated extraction of some of the basic properties of common object formats; the eXtensible Characterisation Languages strive to provide an in-depth description of the complete informational content of an object in an abstract representation [BRH+08b]. These descriptions can be used to derive properties to be measured, and support the automated comparison of these properties when migrating the objects to different formats.

Related to the categorisation of requirements presented above, a distinction can be made between binary criteria which must be fulfilled and gradual factors that need to be balanced against each other. Significant properties of digital objects are most frequently seen as binary criteria that are either preserved or not, and where usually no loss can be tolerated. On the other hand, two preservation actions might both keep all essential characteristics and thus be acceptable. The decision then can take into account gradual factors such as the total costs incurred by each alternative action, processing time, or the assessment of risks that are associated with each alternative. These factors cannot be measured on binary scales. Our approach of tailored utility analysis unifies both kinds of criteria by allowing different scales to be used for the actual measurements of the respective criteria. In the third phase, these measurements are transformed and thus made comparable through the definition of transformation rules, which calculate unified utility values based on the knowledge gained in the experiments. In the final step, critical binary criteria can be used to filter alternatives, while the weighted overall performance across all criteria is then used for the final selection of the best action.

The objective tree thus documents the individual preservation requirements of an institution for a given partially homogeneous collection of objects. The tree as such is entirely independent of the strategy employed, be it migration, emulation, or another [GBR08]. It is of vital importance that it is concerned solely with the *problem space* and does not specify solutions such as *We want to migrate to PDF/A*, unless these decisions have been made already on a higher level, e.g. an institutional policy.

While such specifications are sometimes brought forward in the requirements workshops, they commonly can be traced back to the reasons underlying them, such as preferences for transforming objects to standardised, widely supported file formats and deactivation of active content. The decision to migrate to PDF/A using a specific tool might be the right one; however, without proper documentation of the reasons and the evaluation

[6] `http://hul.harvard.edu/jhove/`

leading to it, the recommendation cannot be considered trustworthy.

The tree shown in Figure 3.6 contains a branch named *technical characteristics*. In this specific case, the institutional policy constrained the class of preservation action to be considered to migration; emulation was not an option. Thus the requirements describe in a very specific form the desired characteristics of the target format the objects should be kept in. These characteristics together form a *risk assessment* of the format and become a central part of evaluating applicable tools and strategies.

A series of case studies have been conducted where objective trees were created for different settings. Examples include electronic publications in a national library [BSN+07]; web archives and electronic documents with national archives [SBNR07]; interactive multimedia in an electronic art museum [BKKR07]; and computer video games [GBR08]. Chapter 5 contains an in-depth discussion on a number of these cases.

Ongoing case studies revise and extend the previously conducted evaluation studies, build concrete preservation plans for specific collections of objects, and cover new scenarios that have not been evaluated yet, such as scientific data or data from personal mobile devices, in a variety of settings.

The experience which is accumulated through carrying out planning activities and requirements definition can be easily shared between institutions through the supporting software, which contains a knowledge base of recurring fragments of objective trees and templates that can be used as a starting point, as described in Section 4.2.2. The knowledge base provides best-practice criteria catalogues that can be applied, further refined and reinserted to the criteria catalogues, providing a feedback loop into the decision process.

The outcome of the first phase is a complete documentation of the planning context, the collection of objects at question, and the specific requirements that form the basis for the evaluation of alternative action paths. Completely specified requirements trees, where every leaf node is assigned a measurable criterion, typically contain between 50 and 150 requirements in about 5 hierarchy levels.

3.4.4 Evaluate alternatives: Define alternatives

The second phase of the planning workflow relies on controlled experimentation. It evaluates potential actions in a quantitative way by applying them to the previously defined sample content and analysing the outcomes with respect to the requirements specified in the objective tree. This empirical evaluation procedure results in an evidence base that underlies the decisions to be taken in the successive phases. It basically provides all information for Section 6 of the preservation plan (cf. Section 3.2.2).

The natural first step of evaluation is to define the possible courses of actions to be taken into consideration. A variety of different strategies may

be applicable; for each alternative action, a complete specification of the entailed steps and the configuration of the software component employed is desired. The discovery of potential actions that are applicable varies in complexity according to the type of content. Often, this implies an extensive search phase, investigating which tools are available to treat the type of objects at hand. Registries holding applicable preservation action components can be consulted for reference and are potentially very beneficial to support the search.

The outcome is a *shortlist* of potential candidates for performing preservation actions, which will be evaluated empirically during the next steps. The description of an alternative includes the tool name and version used, the operating system on which it shall run, and the technical environment specification such as installed libraries and fonts.

3.4.5 Evaluate alternatives: Go/No-Go decision

Before continuing with the experimentation procedure, this step reconsiders the situation at hand and evaluates whether it is feasible and cost-effective to continue the planning procedure. In cases where the evaluation is considered infeasible or too expensive, a reduction of candidate components might be necessary. The evaluation of some tools may also be postponed due to unavailability or cost issues, or because of known bad performance. This is individually described and documented.

3.4.6 Evaluate alternatives: Develop experiment

This step sets up and documents the configuration of the tools on which experiments are carried out, and thus builds the basis for experiment execution in the next step. This includes setup procedures, a documentation of the hard- and software environment, and additional steps needed to carry out the evaluation of experiments, such as setup time measurement and logging facilities.

3.4.7 Evaluate alternatives: Run experiment

In this step, all considered candidate components are applied to the set of sample objects that have been defined in the first phase. This produces a series of experiment results that can be analysed and are stored as evidence. In the case of object conversion, this means that the resulting output files shall be stored for further reference. When evaluating emulators, a documentation detailing the experience of rendering the object is needed. Furthermore, any errors or logging messages occuring are documented, as well as performance issues such as startup and processing time.

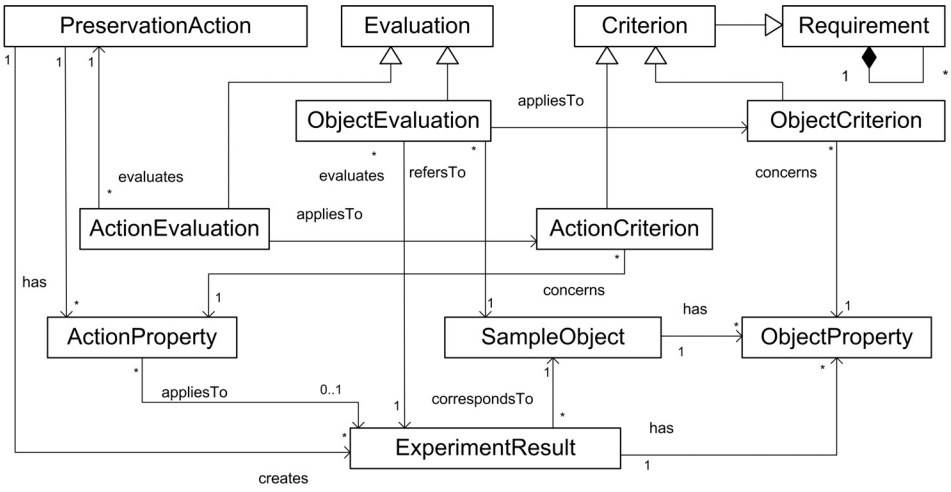


Figure 3.7: Core model of requirements and evaluation

3.4.8 Evaluate alternatives: Evaluate experiment

The evaluation of experiments is based on the requirements specified in the objective tree. All criteria on the leaf level of the objective tree are evaluated, taking into account the empirical evidence resulting from the experiments conducted.

Figure 3.7 shows a simplified abstraction of the core elements of the requirements and evaluation model. Each *Preservation Action* has certain *ActionProperties* and is evaluated through applying it on *Sample Objects* in a controlled experiment. This creates an *Experiment Result* that constitutes part of the evidence base. A *Criterion* is a measurable *Requirement*. It can be concerned with an action (*Action Criterion*) and thus associated with an *ActionProperty*, or with the object an action is applied to (*Object Criterion*). In the latter case, it can be mapped to an *Object Property*. These properties are measured of the original *Sample Object* and the *Experiment Result*, and the obtained values are compared through a comparison metric. Action criteria, on the other hand, are associated with an *Action Property* and evaluated in an *Action Evaluation*.

Thus, the performance of each leaf criterion is measured for each alternative and collected in the objective tree. For some objectives, this has to be done manually, while for others it can be performed automatically using characterisation tools. For example, the previously mentioned criterion *image width unchanged* is an *object criterion* which can be measured by characterisation tools such as JHOVE or XCL and compared automatically for each result of an experiment. Similarly, the relative file size of objects can be measured automatically per object. The relative file size averaged over

the sample objects would then be used as evaluation value for the corresponding criterion. In other cases, information is obtained from registries or inserted manually. For example, the judgement of quality of documentation, or the degree of adoption of a file format, can be queried in registries such as PRONOM, or judged by the preservation planner. Some criteria that are tool-specific rather than object-specific only need to be measured once per alternatice, e.g. the cost of a component. Chapter 6 will discuss automated measurements as data collection means.

Documenting the evaluation of experiment results completes the empirical evidence base for decision making and concludes the second phase of the preservation planning workflow. It has to be noted that the confidence in measurements strongly depends on their reproducibility and level of evidence. For example, exact figures that are consistent through multiple measurements obviously have a much lower uncertainty than subjective judgements without explanatory documentation.

3.4.9 Analyse Results: Transform measured values

The result of the previous phase is an objective tree fully populated with evaluation values for all criteria. However, the measurements in this tree are of varying scales and thus cannot be aggregated and compared directly. In the third phase, the experiment results are consolidated, aggregated, and analysed. As a running example to illustrate the concepts of this phase, we will use the highly simplified weighted example tree in Figure 3.8, which contains three measurable criteria for converting text documents: Correct encoding of characters measured in percent, the orientation of pages measured by a boolean identity, and the speed of conversion, measured in milliseconds.

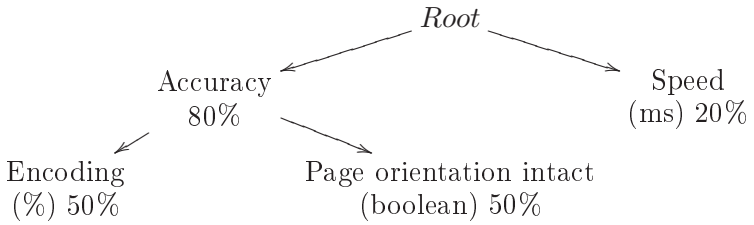


Figure 3.8: Highly simplified requirements tree

In order to apply aggregation and comparison of values over the tree hierarchy, a *utility function* is defined which maps all evaluation values to a uniform target scale of commonly 0 to 5. Hereby, 5 is the optimum value, whereas 0 denotes unacceptable performance that serves as a drop-out criterion.

The utility function can be defined in a variety of ways. For ordinal

Characters > Encoding

Results

Alternatives	1
Adobe Acrobat->DOC	94.0
Convert Doc->DOC	100.0
Adobe Acrobat->HTML	99.0

Transformer

Threshold	Target value
85.0 % of correct characters	-> 1
90.0 % of correct characters	-> 2
95.0 % of correct characters	-> 3
98.0 % of correct characters	-> 4
100.0 % of correct characters	-> 5

Threshold stepping: Steps

Transformed Results

Alternatives	1	Aggregated
Adobe Acrobat->DOC	2	2
Convert Doc->DOC	5	5
Adobe Acrobat->HTML	4	4

Aggregation mode: Arithmetic mean

Figure 3.9: Transformation of evaluation results

values, a mapping is defined for each possible category, resulting in a value between 0 and 5 to be assigned. For a boolean scale, *Yes* might be mapped to 5, whereas *No* will often be mapped to a low value. In this case, a decision has to be made whether the negative result *No* should be acceptable or not, i.e. mapped to 1 or to 0.

For numerical input values, we may rely on a simple linear transformation using threshold settings. Figure 3.9 shows an example transformation setting for the attribute *Character encoding* with percentage values in the range of $[0, 100]$ and the corresponding transformation results. Using simple stepping, the resulting utility function $u(measure)$ in the example case is given in Equation 3.1.

$$u(encoding) = \begin{cases} 0 & \text{if } encoding < 85 \\ 1 & \text{if } 85 <= encoding < 90 \\ 2 & \text{if } 90 <= encoding < 95 \\ 3 & \text{if } 95 <= encoding < 98 \\ 4 & \text{if } 98 <= encoding < 100 \\ 5 & \text{if } encoding = 100 \end{cases} \tag{3.1}$$

More commonly we use piecewise linear interpolation. Other transformations include logarithmic and exponential interpolation. Let v be the evaluation value of a candidate and t_i the list of thresholds in monotonically increasing order with $i = 1, 2, 3, 4, 5$. With the resulting breakpoints (t_i, i), the linear interpolation utility function $u(measure)$ is given in Equation 3.2. For thresholds in decreasing order, the equation is adjusted accordingly.

$$u(measure) = \begin{cases} 0 & \forall v < t_1 \\ i + \frac{v - t_i}{t_{i+1} - t_i} & \forall t_i <= v < t_{i+1} \\ 5 & \forall v >= t_5 \end{cases} \tag{3.2}$$

For both numerical and ordinal values, the definition of *acceptance criteria* is an essential step, where decision makers have to clearly specify the constraints they are willing to accept. This further provides a gap analysis which clearly points out both strengths and limitations of the candidate components under evaluation.

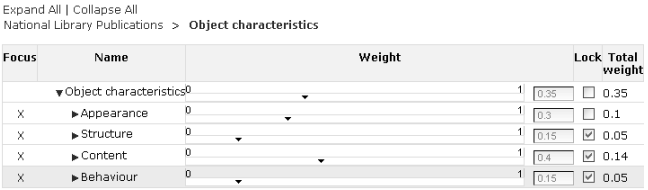


Figure 3.10: Setting importance factors

3.4.10 Analyse Results: Set importance factors

This step takes into account the fact that not all requirements are of equal importance, and assigns weight factors to the nodes in the objective tree.

The weighting of the top-level branches of the requirements trees often depends on institutional policies and may have significant impact on the final evaluation result. In particular, preferences might have to be negotiated between the quality of preservation actions and the costs needed to setup the necessary migration or emulation software, or within the different aspects of significant properties of objects. For example, the 'behaviour' branch of an objective tree for preserving static documents will have a much lower importance weighting than in the context of multimedia art objects, where interactivity is a central aspect. Figure 3.10 shows the supporting software tool Plato balancing the weights of criteria.

The acceptance criteria defined in the transformation rules are used to model the actual utility of the evaluation values, while importance weighting reflects the overall priorities of an institution.

3.4.11 Analyse Results: Analyse results

The final step of the evaluation phase scrutinises the complete evidence base of information produced during the previous phases of the workflow. It analyses the performance of the candidate components in the experiment evaluation to arrive at a conclusion and recommendation for the best component to be employed, and the corresponding configuration.

The measurements described above are transformed and multiplied with the weights of the corresponding requirements. This results in an evaluated objective tree where the leaf criteria have been populated. Aggregating these values leads to a performance value of each alternative action on all levels of the tree hierarchy, which is directly comparable.

Several aggregation methods are supported, of which the most relevant are weighted multiplication and weighted sum. The score of each node n_i is determined by the weighted score of its k children c_j, given that the relative weight of all its children sums up to 1. The score of the leaf nodes is provided by the utility function $u(measure)$ with the measured evaluation value as

Component	**Encoding**	**Orientation**	**Speed**
Component A	98%	*true*	600ms
Component B	100%	*true*	800ms
Component C	100%	*false*	500ms
Component A	4	5	3.6
Component B	5	5	2.8
Component C	5	0	4

Table 3.2: Example evaluation results and utility values

input variable.

For both sum and multiplication, we seek a weighted aggregation function where nodes with a weight of 0 result in a neutral element and the target range is the same as the input range. For weighted sum, this is the well-known linear combination calculating a weighted score as given in Equation 3.3.

$$u^s(n_i) = \sum_{j=1}^{k} w(c_j)u^s(c_j) \tag{3.3}$$

For weighted multiplication, the values are taken to the power of the weight, as given in Equation 3.4.

$$u^m(n_i) = \prod_{j=1}^{k} u^m(c_j)^{w(c_j)} \tag{3.4}$$

The aggregated scoring obtained thereby serves for filtering out candidate components with unacceptable evaluation values – any score of 0 at the criterion level will be reflected as a root score of 0, allowing the decision maker to quickly analyse the root cause of the drop-out. For the remaining alternatives, we generally use the weighted sum aggregation provided in Equation 3.3 to find the candidate component best suited for the given scenario.

We use the transformation function of Equation 3.2 and the thresholds defined in Equation 3.1 to evaluate the tree defined in Figure 3.6. We use piecewise linear interpolation for *speed* and set the thresholds to be {1500,1000,750,500,250} in decreasing order. The loss of page orientation is unacceptable, resulting in Equation 3.5.

$$u(orientation) = \begin{cases} 5 & \text{if true,} \\ 0 & \text{if false.} \end{cases} \tag{3.5}$$

Table 3.2 lists evaluation values and their respective utility. For example, Component A has a measured speed of *600ms*, which translates to a

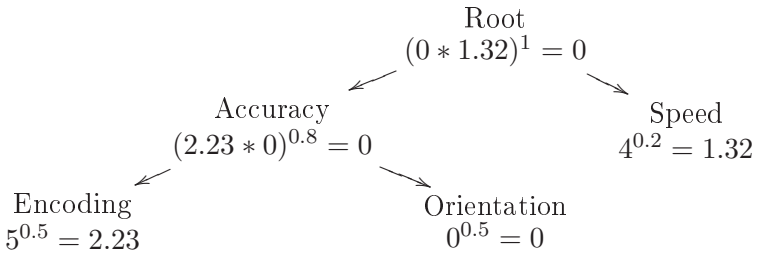


Figure 3.11: Weighted multiplication results for Component C

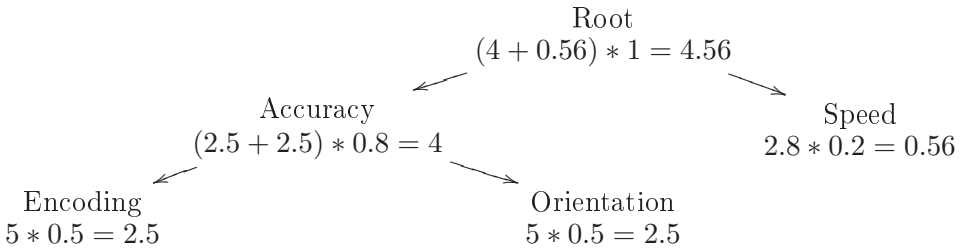


Figure 3.12: Weighted sum results for Component B

utility value of *3.6*. Figures 3.11 and 3.12 illustrate the aggregation of values, showing the weighted aggregated value on each node to illustrate the absolute influence towards the aggregated parent value.

Component C is being eliminated because it breaks the page orientation during conversion, which leads to a utility of 0 for *orientation*, as shown in Figure 3.11. The multiplication has the effect of rejecting the component with a root score of 0, which can be traced back to the criteria responsible for this by following the graph to the leaf level.

Figure 3.12 shows weighted sum results for Component B, which does not exhibit unaccceptable measures. We thus obtain the aggregated weighted result scores provided in Table 3.3. Notice that while Component A is faster than B, the superior evaluation for *encoding* of Component B is weighted stronger and determines the final ranking.

The tree hierarchy can be visualised as shown in Figure 3.13. This

Component	**Weighted multiplication**	**Weighted sum**
Component A	4.28	4.32
Component B	4.45	4.56
Component C	0	2.8

Table 3.3: Aggregated values

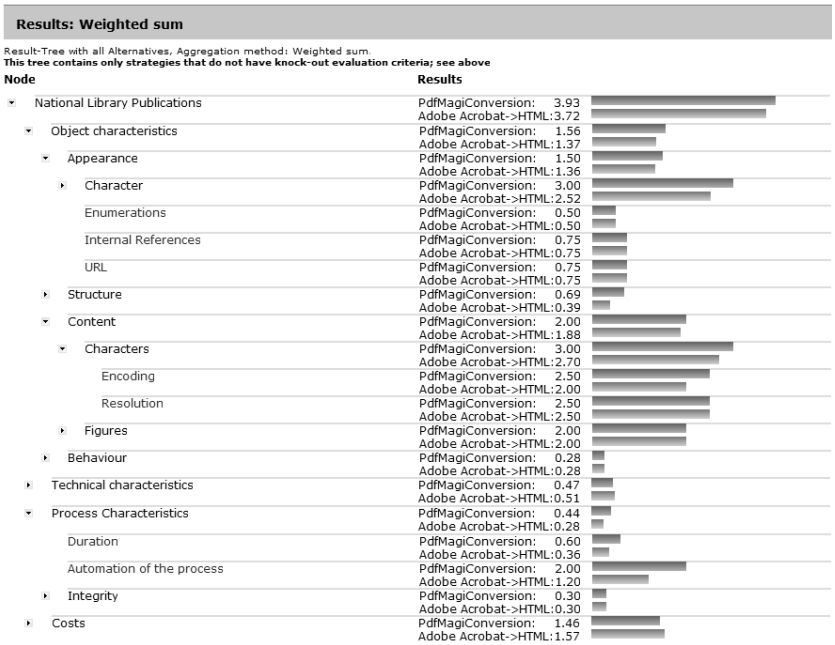


Figure 3.13: Visualisation of results

supports an in-depth analysis of the specific strengths and weaknesses of each candidate component. The definition of acceptance criteria in the utility function further provides a gap analysis which clearly points out both strengths and limitations of candidates on all levels of the hierarchy. This enables evidence-based evaluation and well-informed decision making.

As a result of the evaluation, the preservation planner selects a component to be recommended for integration. The method allows for the selection of multiple components that are considered to be complementary. For example, many conversion tools for electronic documents have problems with entirely preserving the layout as it was displayed in the original environment, whereas migrating a document to an image loses the future potential for full-text search access. In some cases it might be desirable to combine both approaches and thus select multiple components for the incorporation into a preservation system.

As an essential element of the recommendation, the reasons underlying it are documented, together with the expected effects of applying this strategy on the set of objects at hand. For example, it may be known that the easy *editability* of objects will be lost as a direct cause of converting them to a format such as PDF/A. As this might not be a requirement, or not be assigned significant weight, it might not influence the decision in a significant way. However, this reasoning needs to be documented as part of the decision

making procedure.

3.4.12 Build preservation plan: Create executable plan

In the fourth and final phase of the planning workflow, a preservation plan is created, based on the decision for a preservation action. In OAIS terminology this corresponds to the *Develop Packaging Designs & Migration Plans* functionality. This phase specifies a series of concrete actions, along with organisational responsibilities, rules and conditions for executing the preservation action on the collection. This completes the information necessary for the preservation plan as described in Section 3.2.2.

This step of the workflow defines the action steps that form the core part of the preservation plan. This executable action plan includes the triggers for the execution and the conditions under which the preservation action will be carried out, i.e. the preservation component invoked. Hard- and software requirements as well as dependencies on other systems are documented. To enable the execution of the preservation action, tool settings and details about the location of the collection on which the action is to be performed are defined, thus resulting in a *preservation action plan*.

To perform quality assurance of the executed actions, a subset of the criteria used for evaluating solutions can be selected. These criteria should then be evaluated automatically during the execution of the action plan to validate that the defined target ranges of these criteria are met. The necessary documentation that has to be recorded when performing the action is also defined in this step.

For example, if the chosen preservation action to preserve a collection of scanned images is migration to JPEG2000 with a certain component, the action plan specifies exactly the tool and version to be used, the environment on which the component is to be deployed, and any parameter settings. It will also include a specification on metadata events to be recorded. Furthermore, a number of image properties that can be extracted and verified automatically may be specified to include quality assurance during migration.

3.4.13 Build preservation plan: Define plan

While many parts of the preservation planning workflow take care of the technical aspects of the preservation plan, this step mainly defines organisational procedures and responsibilities.

Cost factors influence the decision on a specific alternative. In this step, a more detailed calculation of costs using an approved cost model is performed. Cost models that can be used are for example Life2 [ADM+08] or the TCO model. While an estimate of the costs may be fine for evaluating

Alert	Triggered by OAIS functional entity	Event (examples)
New Collection	Administration	Agreement for a new collection
	Monitor Designated Community	New object type in use
		Frequent submissions of unanticipated formats
Changed Collection Profile	Monitor Designated Community	Use of a new version of an object format in the designated community
		Frequent submission of unanticipated formats or new versions of an object format, or objects with new functionality/characteristics
	Manage System Configuration (in Administration)	Collection grows faster than initially foreseen and specified in the existing preservation plan
Changed Environment	Monitor Technology	Change in the results of the evaluation of objectives of an existing preservation plan, for example price changes or changed risk assessment
		New available preservation strategies, for example new versions of components
		Impending obsolescence of used technology, for example when a target format used in a migration-based preservation plan is becoming obsolete
	Monitor Designated Community	Change of software available at user sites (e.g. indicated by reports about problems with DIPs)
Changed Objective	Monitor Technology	New standards that have to be adopted
	Monitor Designated Community	Change in computer platform or communication technologies used
	Manage System Configuration (in Administration)	Change in designated community of consumers or producer community
		Change of institutional policies
Periodic Review	Develop Packaging Design and Migration Plans	Raised on a scheduled basis defined in the institutional policy or in the preservation plan

Table 3.4: Alerts, triggers and events

the alternatives, the costs for adopting the solution have to be determined as accurately as possible in this step.

The assignment of responsibilities is also documented in this step. Monitoring the process of applying the preservation actions has to be done by a different role than executing the preservation plan. It also has to be monitored if an event occurs that makes it necessary to re-evaluate the plan. Possible triggers for this are a scheduled periodic review, changes in the environment such as newly available tools detected through technology watch, changed objectives (changed target community requirements) or a changed collection profile, when objects show new characteristics diverging from the specified profile.

Examples for these triggers, as well as the OAIS functional entities raising them, are provided in Table 3.4, which lists the alerts, the corresponding triggers and the events firing them. Section 3.5 discusses the relationship to the OAIS functional entities in detail.

Aspects of interest include new versions of object formats that are included in the plan or a change in their risk assessment; changes in the support of technical environments that are used; changes in prices of software tools or services that are used; or a changed availability of tools for preservation action or characterisation. These aspects should be continually monitored after the plan has been specified. Changes might lead to a re-evaluation of potential actions and a potential update of the preservation plan prior to the next periodic review, which should also be scheduled.

Three types of change triggers are defined:

- *Changed Collection Profile.* Changes in the collection profile of an existing collection may require a revision of an established preservation plan. Examples for changes in the collection profile are newly accepted object formats or significant changes in the collection size. An indication for a changed collection profile is also that values measured during the quality assurance deviate significantly from the values measured for the sample objects during the evaluation. It is the responsibility of technology watch functions to ensure that these triggers are actually fired; the corresponding events should then be recorded in the planning documentation.

- *Changed Environment.* The environment of a preservation plan consists of the technical environment, the designated communities and the host institution. Changes in the environment can lead to a change in preferences, for example with respect to the system context in which a preservation action needs to operate. They might also imply a change in factors which influence existing preservation plans, for example changed prices for hardware or software. Other relevant changes are the availability of new preservation strategies or impending obsolescence of object formats which are used in an existing plan. Changes in the environment require a revision of existing preservation plans, while the objectives for the evaluation usually will remain unchanged.

- *Changed Objective.* Changes and developments in the environment can change the objectives for preservation evaluation over time. In this case it is necessary to evaluate existing preservation plans against changed objectives. Examples are changes in high-level policies or legal obligations that have an impact on preferences and objectives, or changes in the designated community, such as the type of software available to the users or new ways of using the objects of interest.

3.4.14 Build preservation plan: Validate plan

In the final stage, the whole documentation on the preservation plan and the decisions taken during the planning process are reviewed. Tests on an extended set of sample objects may be performed in this step to check the validity of the preservation action plan.

Finally, the validated plan has to be approved by the responsible decision maker. Once the plan is approved, no more changes to the plan should be done without formally revising the whole plan.

The final outcome of the four-phase workflow is a completely specified, validated, and formally approved preservation plan defining concrete steps

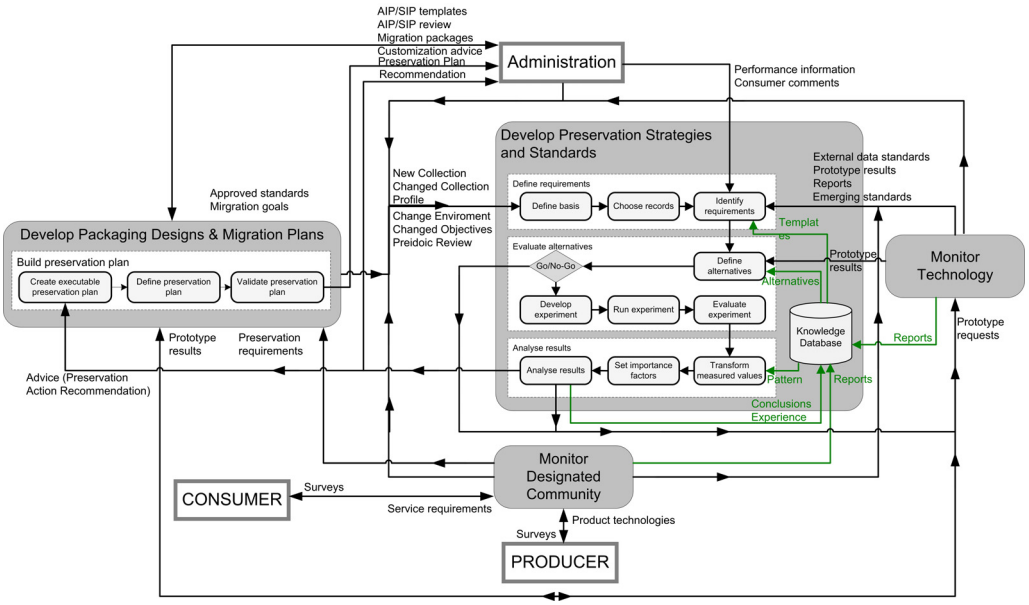


Figure 3.14: Preservation planning in the OAIS model [SR08]

and responsibilities for keeping a certain set of objects alive. The plan includes the complete evidence base of decision making and conforms to the plan definition discussed in Section 3.2.

3.5 Monitoring preservation plans

As with any management activity, *defining the plan* is only the beginning of an iterative lifecycle where plans and operations have to be monitored and revised when changes in the environment require an update to operations. Complementary to the documentation of recorded events that triggered a planning activity, the completed preservation plan also contains a specific definition of events that should trigger a revision of the preservation plan. This enacts a monitoring of those aspects of the environment that are considered to be of particular relevance or particularly prone to change. Evolving environments and changes in the repository will inevitably lead to a need for reaction by adapting plans. This section discusses the question of monitoring operational preservation plans and the environment for changes that require an update. The framework for this discussion is set out by the corresponding watch functions of the OAIS model.

Figure 3.14 shows the integration of the planning approach within the OAIS model and the main information flows. The presented method for defining plans implements the *Develop Preservation Strategies and Stan-*

dards and the *Develop Packaging Designs and Migration Plans* functions. The functional entities of the OAIS model provide possible constraints and requirements for the steps within the planning approach. A detailed analysis of information flows and the planning activities is presented in [SR08].

The *Develop Packaging Designs and Migration Plans* function is responsible for providing detailed migration plans. It uses the recommendation from the function *Develop Preservation Strategies and Standards* as a basis for building a preservation plan, incorporating organisational aspects such as the responsible roles and persons to carry out the plan. It further creates an executable preservation plan that includes mechanisms for quality assurance and capturing metadata.

The *Develop Preservation Strategies and Standards* function is responsible for developing and recommending strategies and standards to preserve the current holdings and new submissions for the future. The first three phases of the planning method evaluate different preservation strategies for a collection or new submissions as described in Section 3.4. The outcome is a preservation action recommendation which identifies the most suitable preservation action for a collection in a specific context. The recommendation is provided to the *Develop Packaging Designs and Migration Plans* function as advice to create a detailed migration plan in Phase 4 of the presented workflow, and to the Administration entity for system evolution.

Staying in the conceptual model of the OAIS, several watch functions provide input for the monitoring of preservation plans.

- The functions *Monitor Designated Community* and *Monitor Technology* perform a watch that provides reports about developments and changes in the designated community and relevant technologies. The function *Monitor Technology* evaluates existing and emerging technologies. This function can be implemented by experiment bases such as the Planets Testbed [AHJ+08] where public experiments are defined and run on benchmark content. For instance, first public benchmark results of new migration components that promise better conversion quality than previously used components can be first indications for closer consideration of these components in the step *Define Alternatives* of the planning method.

  Monitoring technologies also has to cover notifications dealing with risk alerts and format obsolescence. Consider an organisation that has defined a plan specifying that all born-digital photographs in raw camera formats shall be migrated to Adobe Digital Negative (DNG) with a certain tool. If an entry is added in a knowledge base stating that Adobe is dropping support for the DNG format, this raises the risk level of this format and should lead to a notification that the according preservation plans need to be re-evaluated.

- The *Manage System Configuration* and the *Consumer Service* function of the *Administration* entity report performance information of the archiving system and consumer comments to the *Develop Preservation Strategies and Standards* function. These comments can imply requirements regarding access, behaviour and usage of the digital objects in the system. The performance information can thus raise requirements that have to be fulfilled by potential preservation strategies.

  Consider a library that is running a reading room with the rule that 85% of its content should be rendered successfully on 90% of all machines in the reading room. It will need automated tests checking periodically that a certain set of objects that are representative of each part of the entire collection are rendered successfully, applying automated quality measurements. After an OS upgrade, a certain class of objects may fail to render properly. Ideally, this would be detected by an automated measurement framework, for example by using a screen compare function. An event would then be raised and trigger a planning activity.

The functional entities in the OAIS model can trigger new planning activities corresponding to the events defined in Table 3.4. These aspects are not only important during the planning workflow, but also form the basis of an ongoing monitoring process that is essential for successful continuous preservation management.

The deployment of a plan needs to define specific conditions that need to be monitored during execution to ensure ongoing compliance and safe operations. Basically, these service level agreements can be derived from the criteria specified during the requirements stage of the evaluation workflow. The fundamental issue herewith is that the effort needed to evaluate these criteria is substantial if measurements are taken manually. The monitoring is thus only feasible on criteria that can be measured automatically. Chapter 6 will discuss the means to achieve this end.

3.6 Criteria for trustworthy repositories

Trustworthiness is one of the fundamental goals of every repository. This section analyses the Preservation Planning approach in relation to the TRAC checklist [TO07] and the Nestor criteria catalogue [DSS07, nes06]. Both include, among others, several criteria covering the following aspects:

1. Procedures, policies, and transparent documentation;
2. Monitoring, evolvement, and history of changes;
3. Significant properties and information integrity.

Aspect	Criterion	Artefacts and actions
Procedures and policies	TRAC A3.1 Repository has defined its designated community(ies) and associated knowledge base(s) and has publicly accessible definitions and policies in place to dictate how its preservation service requirements will be met. Nestor 4.4 The digital repository engages in long-term planning Nestor 8. The digital repository has a strategic plan for its technical preservation measures TRAC A3.2 Repository has procedures and policies in place, and mechanisms for their review, update, and development as the repository grows and as technology and community practice evolve.	The preservation plan specifies monitoring conditions and triggers. Periodic reviews following the specified workflow lead to revisions of the plan. The set of preservation plans define how the repository policies are fulfilled and verifiably document how the service requirements are met.
Transparency and documentation	TRAC A3.6 Repository has a documented history of the changes to its operations, procedures, software, and hardware. TRAC A3.7 Repository commits to transparency and accountability in all actions supporting the operation and management of the repository, especially those that affect the preservation of digital content over time. TRAC B3.1 Repository has documented preservation strategies. TRAC B3.4 Repository can provide evidence of the effectiveness of its preservation planning.	The preservation plan contains a change history. The preservation plan is fully documented and traceable. All evidence from the experiments is kept as inherent component of the plan. Empirical evidence obtained through controlled experimentation provides an indication of effectiveness, but long-term studies are needed for validation.
Monitoring	TRAC B3.2 Repository has mechanisms in place for monitoring and notification when Representation Information (including formats) approaches obsolescence or is no longer viable. Nestor 5.3 The digital repository reacts to substantial changes TRAC B3.3 Repository has mechanisms to change its preservation plans as a result of its monitoring activities.	As part of the preservation plan, appropriate monitoring conditions are specified. Triggers result in a planning cycle, potentially revising the plan.
Periodic reviews	TRAC A3.4 Repository is committed to formal, periodic review and assessment to ensure responsiveness to technological developments and evolving requirements.	Environment conditions to monitor are specified, periodic reviews following the planning workflow are conducted.
Significant properties	TRAC B1.1 Repository identifies properties or information content it will preserve for digital objects. TRAC A3.8 Repository commits to defining, collecting, tracking, and providing, on demand, its information integrity measurements. Nestor 9.2 The digital repository identifies which characteristics of the digital objects are significant for information preservation.	The objective tree provides a full specification of all properties considered to be significant, each of which is linked to corresponding measurements as appropriate. The automatically measurable subset of these can be collected and tracked during automated plan execution.

Table 3.5: Supported criteria for trustworthy repositories

The next paragraphs discuss each of these aspects, while Table 3.5 contains a list of specific criteria relevant to each aspect and summarises which artefacts and actions of the planning approach contribute to the fulfilment of each criterion. DRAMBORA, on the other hand, can be applied to analyse and verify the risks that apply to preservation planning activities within an organisation and can thus support the ongoing improvement and implementation within an organisation.

3.6.1 Procedures, policies, and transparent documentation

Well-defined policies and transparent documentation are considered essential by both TRAC and nestor. The TRAC report states that *'transparency is the best assurance that the repository operates in accordance with accepted standards and practice. Transparency is essential to accountability, and both are achieved through active, ongoing documentation.'*[TO07]

The Preservation Planning approach evaluates preservation strategies in a consistent manner, enabling informed and well-documented decisions. It enforces the explicit definition of preservation requirements in the form of specific criteria. The definition of the criteria and their measurement units as well as the evaluation itself have to be as objective and traceable as possible. This complete specification of underlying policies and constraints, the collection profile, the requirements and evaluation results as well as the resulting preservation plan result in a comprehensive documentation and a reliable, accountable and transparent decision on which strategies and components to deploy.

The software tool *Plato*, which implements the planning approach, supports automated documentation of the planning activities. The potential effects of preservation strategies on digital objects are evaluated, and the history of preservation plans created, reviewed and updated with the planning method documents the operations, procedures, software, and hardware used in the context of preservation actions. Additional documentation, of course, needs to be provided for the general system hardware and procedures outside the preservation planning setting.

3.6.2 Monitoring and change management

The institutional policies need to define watch services for the collection and its development and for changes in technology and the designated communities. These watch services trigger the according alerts as defined in Section 3.2.2 and Table 3.4. By reviewing the affected plans using the planning workflow, the repository is able to assess the impact of changes and react accordingly. The review verifies an implemented preservation plan, considering changes in requirements and practice or changes in the collection, and might result in an update of the preservation plan, replacing the existing plan. It supports impact assessment and reaction as environments and technology change. The accumulated history of changes and updates to preservation plans is fully documented and provides a traceable chain of evidence.

3.6.3 Significant properties and information integrity

The objective tree specifies requirements and goals for preservation solutions. The core part of it is formed by the specification of significant properties. These requirements document the properties of objects that have to be preserved, and align them with automated measurement tools, if these are available. An automatically measurable subset of these can be collected and tracked during automated plan execution to verify information integrity. The supporting tool described in Chapter 4 provides templates and fragment trees to facilitate the tree creation. These templates and fragments are based

on experiences from case studies and enable the exchange of best practice.

3.7 Summary

This section introduced a systematic approach to defining, creating and monitoring preservation plans in a repeatable, transparent and traceable way. We first outlined the main requirements of the scenario and defined the structure of a preseration plan as follows.

> A preservation plan defines a series of preservation actions to be taken by a responsible institution due to an identified risk for a given set of digital objects or records (called collection). The Preservation Plan takes into account the preservation policies, legal obligations, organisational and technical constraints, user requirements and preservation goals and describes the preservation context, the evaluated preservation strategies and the resulting decision for one strategy, including the reasoning for the decision. It also specifies a series of steps or actions (called *preservation action plan*) along with responsibilities and rules and conditions for execution on the collection. Provided that the actions and their deployment as well as the technical environment allow it, this action plan is an executable workflow definition [BKG+09].

The key issues in planning are selecting the right action, defining the plan around it, and monitoring the execution of the plan. We thus reformulated the problem of evaluating preservation components and highlighted its main characteristics. An analysis of existing component evaluation and selection approaches showed that none of the existing approaches is fully applicable. We thus presented a component evaluation and selection framework geared at automated measurements, and showed in detail how it can be implemented for preservation planning.

The next chapter will present a planning tool developed to implement and support this framework. We will then in Chapter 5 describe a series of case studies and use them to analyse experiences in applying the described approach to real-world problems.

Chapter 4

Plato: The Planning Tool

The last chapters have outlined the issues surrounding preservation planning and presented a framework for creating plans. Until now, preservation planning is largely a manual and tedious process where available solutions are evaluated against the specific requirements of a particular situation. This chapter shows how this method is put into practice. It describes the architecture and features of a decision support system for preservation planning based on a service oriented approach for distributed preservation solutions. We describe an architecture of preservation services with the planning tool Plato as its core. We outline the main features, present the overall integration architecture and highlight examples of service integration. The resulting planning environment significantly improves automation and provides sophisticated tool support for decision making.

4.1 Overview

The planning tool *Plato* implements the preservation planning methodology described in the last chapter and integrates registries and services for preservation action and characterisation. The tool enables preservation planners to create plans conforming to the structure outlined in Section 3.2. It provides substantial automation and guidance, and it documents all decisions made in the planning process. It furthermore provides a sophisticated web-based interface for guiding the planner through the process.

The first public version of Plato has been published online in 2008; since then, the number of users has grown steadily and exceeded 500 in March 2010. The system shared the award for *Best Demo* at ECDL 2008 with the search engine Summa. It is publicly accessible online[1]. Two short papers give a quick introduction to the main features of the tool [BKRH08, BKR10].

Plato is a J2EE-based web application relying on open frameworks such

[1] `http://www.ifs.tuwien.ac.at/dp/plato`

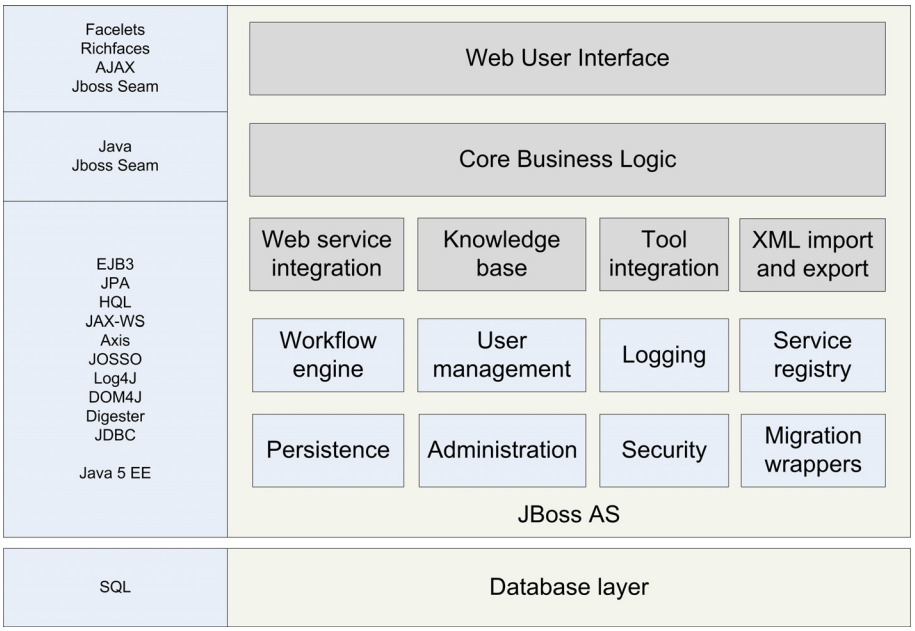


Figure 4.1: Plato layered architecture

as Java Server Faces[2], Facelets[3] and AJAX[4] for the presentation layer, Enterprise JavaBeans[5] for the backend, and JBoss Seam[6] for integrating backend and frontend, as outlined in Figure 4.1.

The tool is integrated in an interoperability framework that supports loose coupling of services and registries through standard interfaces and provides common services such as user management, security, and a common workspace. Based on this technical foundation, the aim is to create an interactive and highly supportive software environment that advances the insight of preservation planners and enables proactive preservation planning.

Figure 4.2 shows the home screen of the Planning Tool. The main elements of the screen are the following.

- Introductory information and extensive background documentation on the planning approach, the workflow, and how to use the planning tool;
- Action links for listing plans belonging to the user as well as published plans that have been made available by other users for reference;

[2] http://java.sun.com/javaee/javaserverfaces
[3] https://facelets.dev.java.net/
[4] http://www.jboss.org/richfaces
[5] http://java.sun.com/products/ejb/
[6] http://seamframework.org/

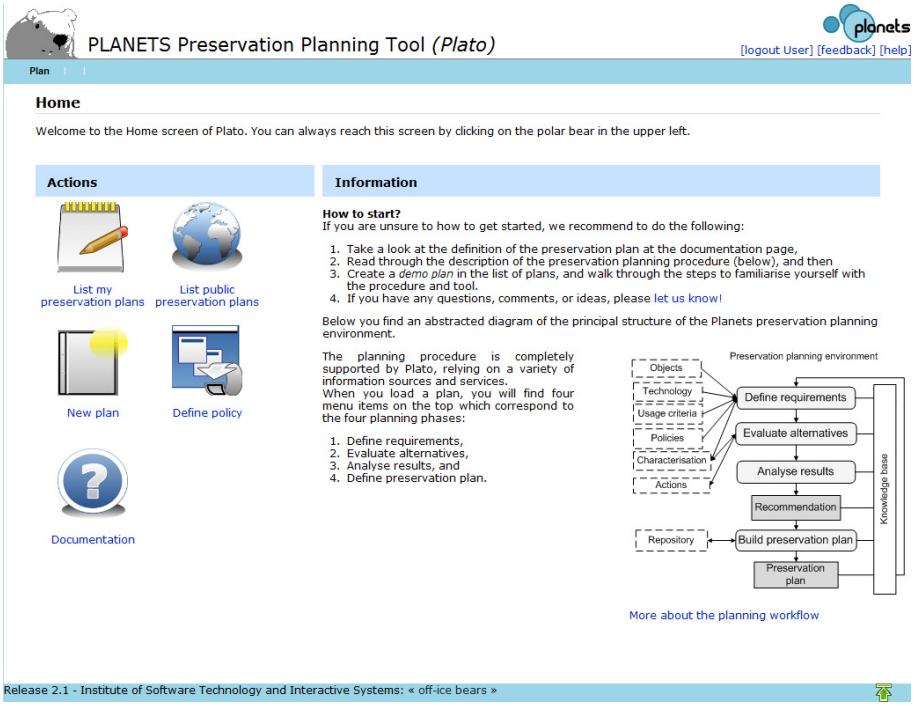


Figure 4.2: Plato home screen

- Action links for documenting an organisational policy, and for creating a new preservation plan; and
- The option to provide comments, bugreports and general feedback to the core development team.

The main benefits of Plato are guidance, automation, and documentation. Preservation action services are discovered in registries and invoked through a flexible layer of adaptors. The time-consuming and inherently subjective process of evaluating the results is being objectified and automated as far as possible by mapping identified requirements such as essential characteristics of objects to properties that can be automatically extracted and compared by characterisation tools. This will be discussed in detail in Chapter 6. A knowledge base supports the preservation planner step by step in identifying requirements and mappings to characteristics as well as transformation of the results and importance weighting of the requirements.

4.2 Workflow support

The planning tool implements the entire workflow as described in Section 3.4. Figure 4.3 repeats the planning environment shown earlier. In princi-

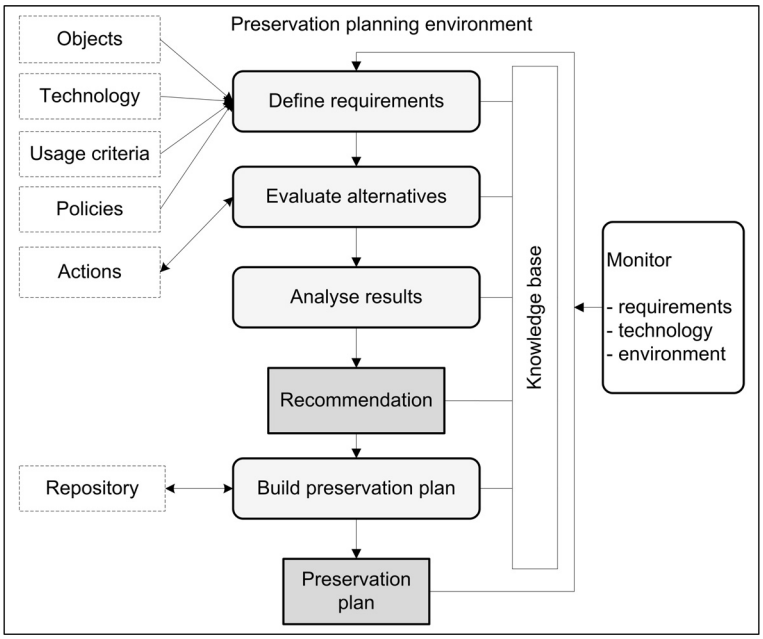


Figure 4.3: Preservation planning environment

ple, there are three primary integration aspects to consider: (1) Integrating registries for information discovery; (2) Integrating services for preservation action and measurements; and (3) Proactively supporting the planning with a knowledge base that holds reusable patterns and templates for requirements recurring in different planning situations.

Figure 4.4 illustrates the sequence of screens that comprise the 14-step planning workflow.

1. In *Define Basis*, the planner documents all fundamental constraints and describes the main cornerstones of the planning procedure. Applying policies associated with the organisation and defined in the knowledge base are connected and documented automatically.

2. In *Choose samples*, the user uploads the representative set of sample objects through the browser, so that the data are stored on the server. These files are characterised automatically and form the basis for empirical evaluation later.

3. In *Identify requirements*, the tool provides a sophisticated tree editor for creating the objective tree, integrating closely with mind-mapping software that provides intuitive editing features.

4. In *Define alternatives*, a number of registries are queried for candidate components that are applicable to the sample content uploaded before,

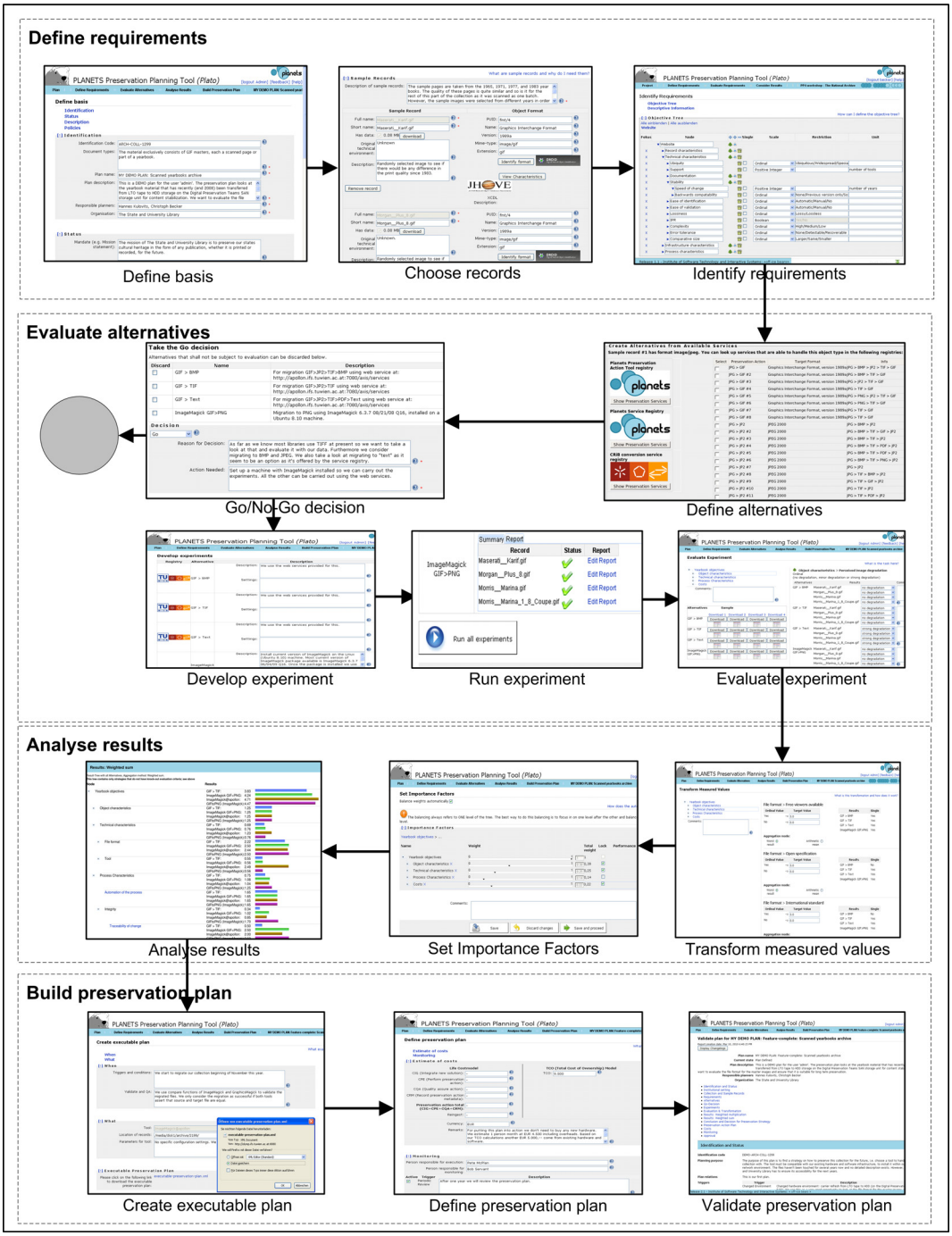


Figure 4.4: Workflow steps in Plato

and the user can decide which to include in the evaluation procedure.

5. The *Go/No-Go decision* is recorded and documented; without choosing to continue the experiment, the rest of the workflow is not accessible.

6. In *Develop experiment*, the planner documents desired parameter settings for the experiment execution.

7. In *Run experiment*, all chosen candidate components are applied to all sample objects, and the results are stored as evidence base for subsequent analysis. Much of the experiment runs is automated by web services, where the user only has to push a *play* button to carry out conversion procedures and access emulation environments.

8. In *Evaluate experiment*, all criteria are evaluated taking into account the evidence generated in the experiments. This can entail complex evaluation procedures and traditionally involves substantial human effort. Chapter 6 will present improvements in the degree of automation.

9. In *Transform measured values*, the planner defines the utility functions for all criteria, based on a knowledge of realised measurements.

10. In *Set importance factors*, the user can set the relative weights of criteria supported by an intuitive mechanism that balances weights automatically.

11. In *Analyse results*, the tool provides a flexible graphical visualisation of the performance of all components, linked to detailed measurement and evaluation reports. The planner then chooses the component to recommend for deployment, and documents the reasoning.

12. In *Create executable plan*, the action steps to be taken are defined based on the taken recommendation. Depending on the nature of the chosen component, the planning tool can often generate an executable action specification in XML that can be readily deployed on a compatible preservation infrastructure.

13. In *Define preservation plan*, roles and responsibilities as well as cost indications and monitoring conditions are documented.

14. Finally, in *Validate plan*, the responsible planner signs off the finished and completed preservation plan to be put into action. The plan is then frozen and cannot be changed unless formally revised by an authorised planner.

The next sections highlight some of the aspects of tool support:

1. Sample objects,

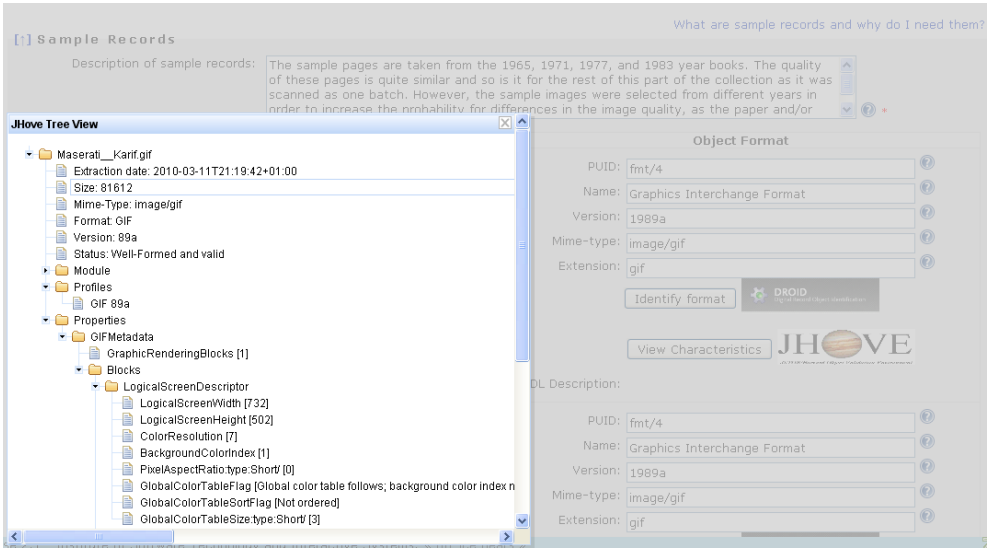


Figure 4.5: Plato showing object properties extracted by JHOVE

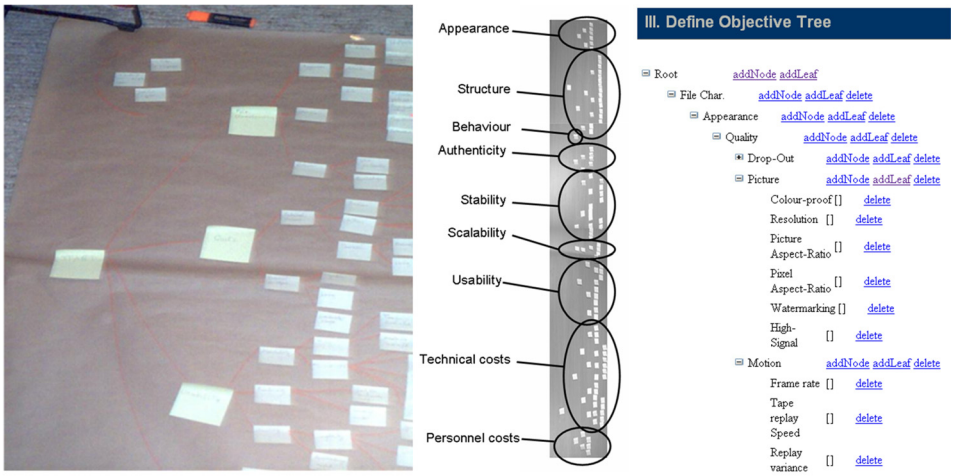
2. Requirements definition,
3. Experiment execution and evaluation,
4. Visual analysis of results, and
5. Preservation plan definition.

4.2.1 Sample objects

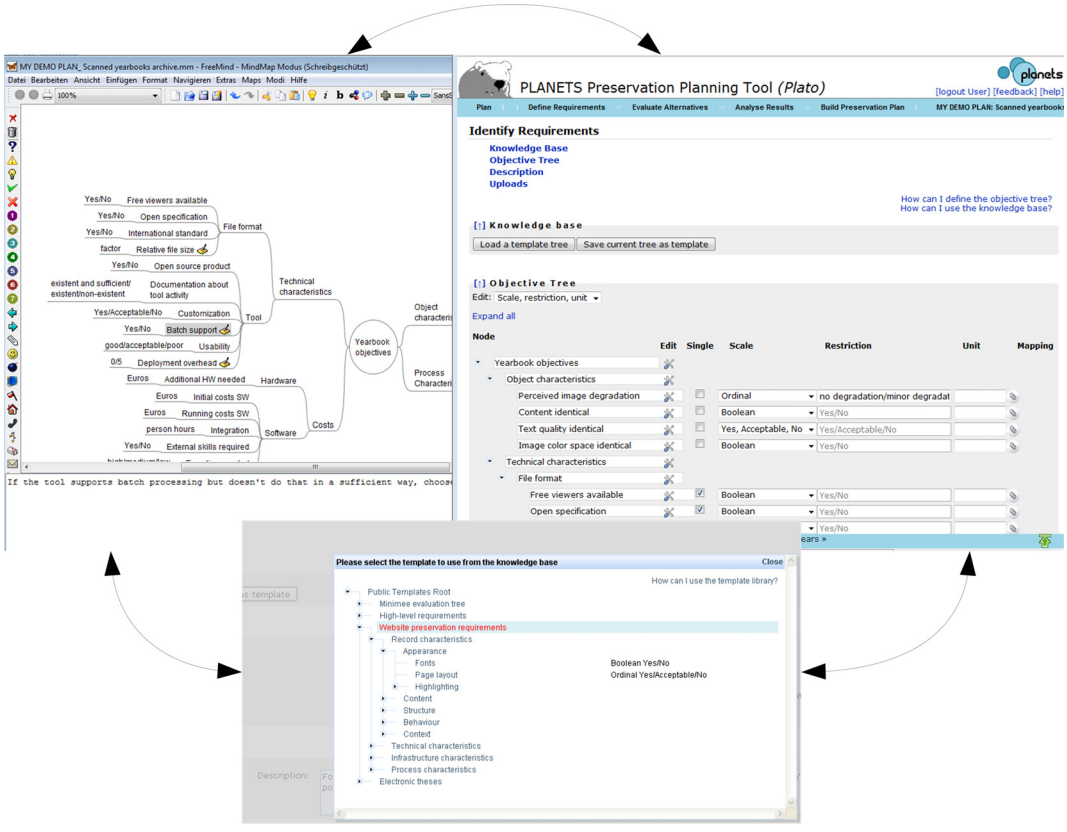
Sample objects are uploaded into the planning tool and stored as integral part of the planning procedure, providing the test data for empirical evaluation. They are thus characterised in depth at the point of deposit in the planning tool, using a combination of state-of-the-art characterisation tools such as DROID[7], the Digital Record Object Identification tool, and JHOVE[8], the JSTOR/Harvard Object Validation Environment. For some content types, the full information model is extracted using the XCL tool-suite described in Section 2.6.

Figure 4.5 shows the display of a sample object's properties extracted by JHOVE in a tree view. These properties are later compared to the properties of the objects that result from the application of candidate components to the sample set.

[7] `http://sourceforge.net/projects/droid/`
[8] `http://hul.harvard.edu/jhove/`



(a) Requirements definition before Plato



(b) Requirements definition in Freemind and Plato

Figure 4.6: Requirements definition: From analog to digital.

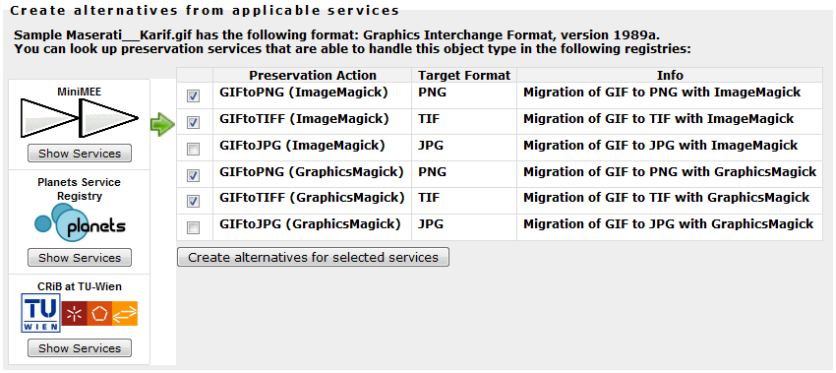


Figure 4.7: Plato listing migration services for GIF images

4.2.2 Requirements definition

The specification of requirements in a tree structure is often done in a workshop setting. Traditionally, participants of these workshops were using sticky notes on paper to collect their requirements; the objective tree was then documented by an expert. Figure 4.6(a) shows photographs of such objective trees side by side with a screenshot of the DELOS Testbed.

In Plato, this step is supported by both a flexible web interface and a direct tree export and import to and from mind-mapping software[9], supporting round-trip two-way editing as depicted in Figure 4.6(b). Users can download the tree and work in the mind-mapping tool, then re-upload the tree and continue editing in the browser, where they have access to a growing library of fragments and template trees. This knowledge base can be used to store recurring tree fragments, such as weighted factors for assessing format risks, significant property descriptions for certain types of objects, or common usage requirements. It also stores measurable properties and allows users to align these with their requirements to facilitate automated measurements as described in the next chapter.

Usually, the initial requirements elicitation from the stakeholders is supported by mindmapping software, while the fine-tuning and specification of measurements is carried out directly in the web interface of the planning tool.

4.2.3 Experiments execution and evaluation

Defining and evaluating experiments where preservation components are tested for their suitability in a given scenario is a complex procedure. The planning tool provides ample support in the aspects of discovery, invoca-

[9] `http://freemind.sourceforge.net`

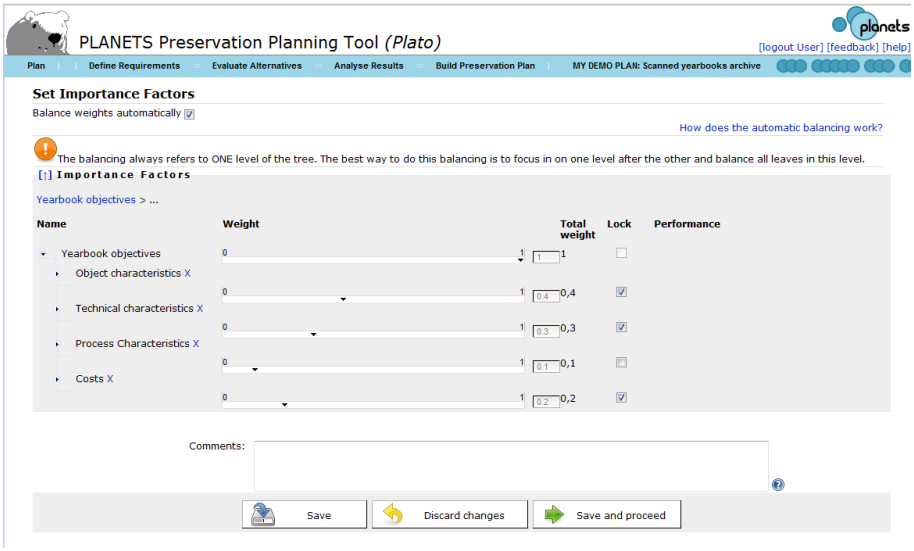


Figure 4.8: Plato balancing importance factors

tion, and measurements, by integrating a number of services and providing integrated access.

Figure 4.7 shows the discovery of potential preservation action components in the step *Define alternatives*. Based on the type of objects that have been defined previously, a number of registries is queried and the set of candidates is filtered according to applicability. The user can then include components as desired, and complement the list of obtained alternatives by manually exending it.

During experiment execution, the components that have been obtained automatically can usually be invoked directly by the planning tool, without the need to install them separately on a dedicated server as previously necessary. Instead, the planner initiates the experiment, and the planning tool invokes all candidates, relying on their web service interfaces. The tool uses a flexible set of adaptors to invoke different web services, hiding the variations of the interfaces from the user. Plato then collects all results, uses the previously mentioned characterisation tools to describe the results' properties, and stores the entire evidence base for subsequent analysis. We will discuss the architecture required to support this automation in Section 4.3.

4.2.4 Visual analysis of results

In the phase of analysing results, Plato supports the specification of importance factors by balancing relative weights on each level as shown in Figure 4.8. Analysis of results is then facilitated by a dynamic and flexible visualisation as depicted in Figure 4.9, where the planner can choose between

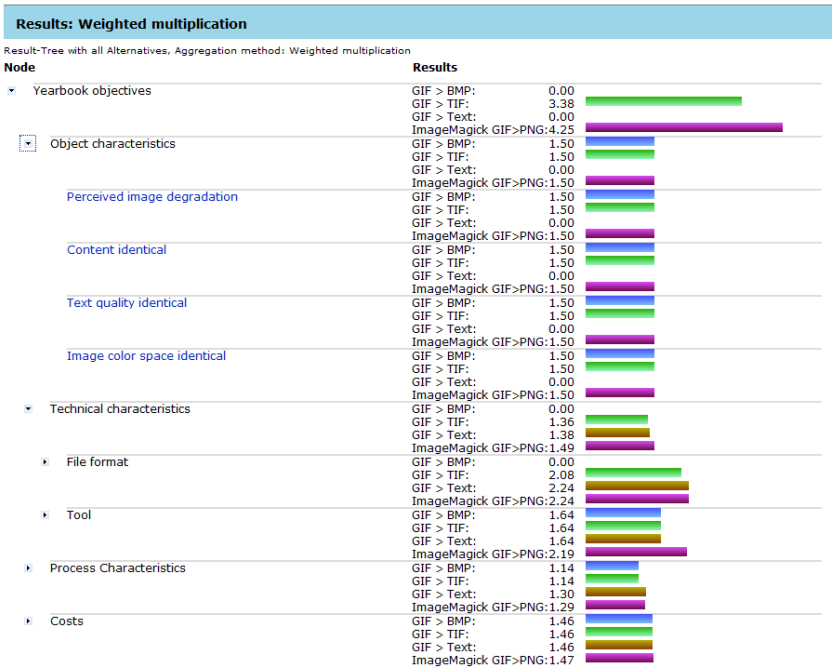


Figure 4.9: Visualisation of results in Plato

different aggregation methods and dynamically configure the displayed information content to analyse the strengths and weaknesses of the alternatives considered. By navigating down the tree hierarchy to the measurable criteria level, the planner can trace down the reasons for certain evaluation results and get in-depth information from the empirical evidence base. At the leaf level, all measured values as well as the utility functions and the resulting utility values are linked and directly accessible. Based on the analysis, a well-documented and solid recommendation for a solution can be made.

4.2.5 Preservation plan definition

The recommended action component then forms the core of a concrete preservation action plan. Depending on the selected component, this can be an executable preservation action workflow as described in [KSJ+09, SKS+09]. In this case, an executable XML file is generated by the planning tool and can be deployed in such an environment.

The plan also specifies the person responsible for approval and documents monitoring conditions corresponding to the triggers defined in Section 3.2. For example, file format risk assessment scores will need to be monitored to react if, for instance, a format defined as target in the implemented plan is becoming obsolete.

PLANETS Preservation Planning Tool *(Plato)*

[logout User] [feedback] [help]

Plan | | MY DEMO PLAN: Scanned yearbooks archive

The plan you loaded has reached the state Plan Validated. Therefore you have been directed to the subsequent workflow step.

Validate plan for MY DEMO PLAN: Scanned yearbooks archive

Report creation date: Feb 1, 2010 11:27:13 AM

Display Changelogs

Plan name MY DEMO PLAN: Scanned yearbooks archive
Current state Plan Validated
Plan description This is a DEMO plan for the user 'user'. The preservation plan looks at the yearbook material that has recently (end 2008) been transferred from LTO tape to HDD storage on the Digital Preservation Teams SAN storage unit for content stabilization. We want to evaluate the file format for the master images and ensure that it is suitable for long term preservation.
Responsible planners Hannes Kulovits, Christoph Becker
Organization The State and University Library

- Identification and Status
- Institutional setting
- Collection and Sample Records
- Requirements
- Alternatives
- Go-Decision
- Experiments
- Evaluation & Transformation
- Results: Weighted multiplication
- Results: Weighted sum
- Conclusion and Decision for Preservation Strategy
- Preservation Action Plan
- Costs
- Monitoring
- Approval

Identification and Status

Identification code ARCH-COLL-1299
Planning purpose The purpose of this plan is to find a strategy on how to preserve this collection for the future, i.e. choose a tool to handle our collection with. The tool must be compatible with our existing hardware and software infrastructure, to install it within our server and network environment. The files haven't been touched for several years now and no detailed description exists. However, The State and University Library has to ensure its accessibility for the next years. Rescanning the magazines is not an option as some of the pages don't exist anymore. So some parts exist in digital form only.
Plan relations This is our first plan.
Triggers

Trigger	Description
Changed Environment	Changed hardware environment: carrier refresh from LTO tape to HDD (on the Digital Preservation Team SAN). We see this as a very good opportunity to look at the file format for the master images and ensure that it is suitable for long term

Release 2.1 - Institute of Software Technology and Interactive Systems: « off-ice bears »

[↑] Policies

Expand all

- PP/2 Policy requirements

Institutional setting

Document types The material exclusively consists of GIF masters, each a scanned page or part of a yearbook.
Mandate The mission of The State and University Library is to preserve our states cultural heritage in the form of any publication, whether it is printed or recorded, for the future.
Designated community General public.
Applying policies See policy model.
Relevant organisational procedures and workflows Library account is needed for access.
Contracts and agreements specifying preservation rights Copyright held for the physical material. Legal mandate implies that transforming logical representation of the content is allowed.
Reference to agreements of maintenance and access None.

Collection and Sample Records

Samples description: The sample pages are taken from the 1965, 1971, 1977, and 1983 year books. The quality of these pages is quite similar and so is it for the rest of this part of the collection as it was scanned as one batch. However, the sample images were selected from different years in order to increase the probability for differences in the image quality, as the paper and/or print quality of the magazines may have changed over the years.

Collection profile

Collection ID: Yearbook-collection-TSL-1200
Description: The first part of the yearbook collection of the Danish car periodical "Bil-Revyen". This part contains the yearbooks published in the years 1965-1989.
Type of objects: This part of the collection consists of GIF files.
Number of objects: 9000
Expected growth rate: No magazines have been scanned since 2006 and when the scanning is resumed then they will be scanned directly to the new preservation format. Thus, the future growth of one magazine per year will not be related to this preservation plan.

Name	Short name	Description	Original environment	Data	Object-format
Maserati__Karif.gif	Maserati__Karif.gif	Randomly selected image to see if there would be any difference in the print quality since 1983.	Unknown	Data existent (81612B)	**PUID:** fmt/4 **Name:** Graphics Interchange Format **Version:** 1989a **mime-type:** image/gif

Figure 4.10: First part of a preservation plan in Plato

Figure 4.10 shows the first part of a plan ready for sign-off. Once the plan is approved, it cannot be changed unless it is formally revised by an authorised user.

The entire preservation plan can be exported to an XML file that includes the complete evidence base – i.e. the sample objects, the documented scenario and applying constraints and policies, the candidate components considered, the results of their experimental evaluation and the analysis documentation. The preservation plan can then be archived in the digital repository system for reference and documentation purposes. It can further be uploaded into any different installation of the planning tool, which provides backward compatibility to previous versions of the XML schema as described in [Kra09].

4.3 Integration architecture

While the preservation planning approach and the supporting planning tool outlined above provide considerable support and guidance, we need a link to existing tools and services performing preservation action and characterisation as well as a dynamic integration of information from different, partly heterogeneous information sources (registries). This section outlines an integration architecture and a pluggable framework for automating the evaluation of preservation actions in the described context.

Distributed preservation infrastructures need registries that hold up-to-date information on aspects such as object formats, available characterisation services, and applicable preservation action services. Format registries such as PRONOM [BCH+07] or the Global Digital Format Registry (GDFR) partly cover this need. However, none of them currently contains references to preservation actions and web services. Plato is thus able to dynamically load additional registries as configured and connect to various registries containing components with different interfaces through a unifying adaptation layer that allows transparent invocation and comparison of components.

Figure 4.11 shows the overall building blocks of the integration architecture. The two fundamental aspects are integration of action components, and characterisation and evaluation. The knowledge base integrated in Plato contains quality models and measurement criteria. Repository planning adaptors are needed to interface to repository systems such as DSpace[10], RODA[11], and ePrints[12]. We are currently developing such interfaces, but will not describe them in this thesis.

Component integration is needed for accessing (remote) preservation action components and services that come in different flavours and varying

[10] http://www.dspace.org/

[11] http://www.fedora-commons.org/about/examples/roda

[12] http://www.eprints.org/

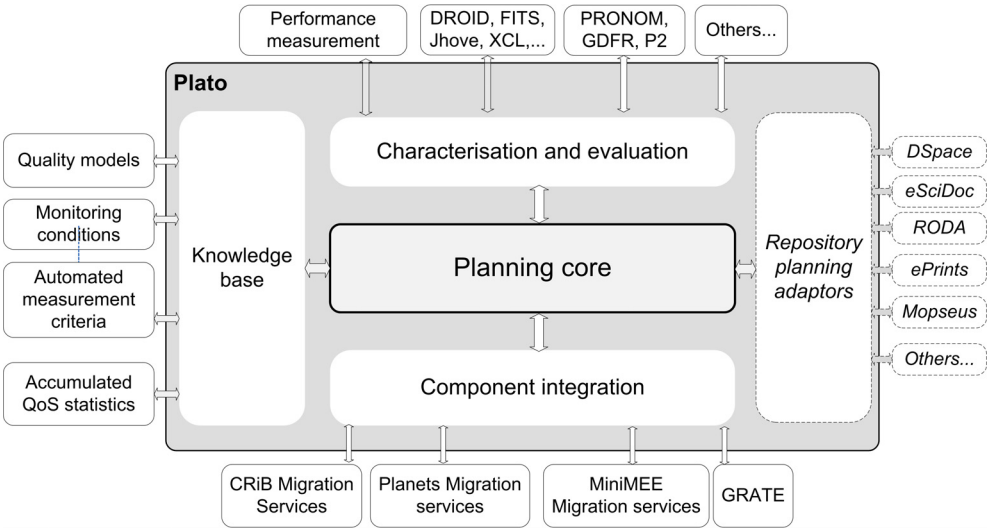


Figure 4.11: Overall integration architecture

form. A number of migration services are available online that convert objects [BFK+08]. On the other hand, emulators can be a viable alternative in certain instances. Remote access to emulation can support the evaluation and the decision whether or not the additional effort for setting up an emulation environment is both feasible and valuable in a given planning situation.

Characterisation and evaluation relies on querying information sources and accessing analysis tools. *Registry adaptors* provide access to information sources. This primarily refers to registries holding information about preservation action tools and services, but also includes access to preservation characterisation registries that hold information such as risks of file formats. *Characterisation adaptors* access tools and services which can identify file formats, assess the risks of digital objects, extract some or all of their properties and compare these, and extract other metadata required for evaluation. The characteristics extracted by characterisation tools and services can be of considerable heterogeneity and complexity. Moreover, the tools are rapidly evolving. We thus rely on an extensible architecture for the automated evaluation of objectives and criteria leveraging these services.

Plato integrates an array of services from different sources, as described in [BFK+08, BKK+09b]. The next sections describe the integration of preservation action components for migration and emulation, and the integration of characterisation tools. Chapter 6 will build on this infrastructure and discuss the more complex issues of measurement in detail.

4.3.1 Migration

Plato currently integrates three types of migration registries and services: The Planets suite, CRiB, and MiniMEE, a minimal migration and emulation engine. We will discuss each of them shortly.

Name	Description
AbiWord	Converter for DOC, HTML, PDF, RTF, TXT, ODT
AviDemux	MPEG to AVI
Dia	Image converter
ffMPEG	Image converter
Ghostcript	PS to PDF Migration
Gimp	Image converter
GraphicsMagick	Image converter
ImageMagick	Image converter
InkScape	SVG TO PS/EPS/PDF/PNG converter
Jasper19	JP2 to JPEG
Java-SE	A wrapper for the migrations supported by the Java SE built-in ImageIO library.
JJ2000	Image converter PPM to JP2
JTidy	HTML to HTML
Mdb2SIARD	MDB to SIARD
MsgText	Extracts Text and attachements from .msg mails
NetPBM	Image converter
OpenJpeg	Tiff to JP2 converter
Pdf2PdfAMayComputer	PDF to PDF/A converter
Pdf2Ps	PDF to PS converter
PdfBox	PDF to HTML 4.0 / UTF-16
SanselanMigrate	Image converter (pure Java)
SoX	Audio converter (WAV, AIFF, FLAG, OGG, RAW)
Xena	Convertsion of audio, databases, documents, email, graphics
DioscuriArjMigration	Convert arj archives to self-extracting .exe files. The service is a wrapper for the original MS-DOS Arj-Tool running on Dioscuri[13].
DioscuriPnmToPngMigration	Converter between PNM (bitmaps) and PNG running on Dioscuri

Table 4.1: Migration services available in the Planets framework

Planets

One of the central goals of the Planets Interoperability Framework[SKS+09] (IF) is the provision of preservation action and specifically migration components through a common, distributed infrastructure, to enable experimentation and evaluation. The IF contains a service registry that holds web service descriptors of migration components wrapped in a uniform interface. Table 4.1 lists the migration components that have been integrated into the framework. Most of them are standard converters, but several are of spe-

Images	Documents
GIF2JP2, PNG2TIF, BMP2JP2, BMP2JPG, TIF2GIF, TIF2PDF3, TIF2PDF2, TIF2PDF, TIF2JP2, JPG2BMP, JP22TIF, PNG2JP2, JPG2PDF, JPG2MultipageTIF, JPG2TIF2, TIF2JPG, TIF2PNG, GIF2TIF, BMP2TIF, JPG2TIF, JPG2PNG, TIF2BMP, PNG2JPG, JPG2JP2, MultipageTIF2JP2,	PDF2Text, PDF2TextLayout, ODT2DOC97, PDF2JPG, ODT2DOC, RTF2ODT, ODT2PDF, ODT2RTF, PDF2JP2, DOC2ODT, ODT2TEXT, PDF2TIF, DOC2PDFOOWRITER, PDF2MultipageTIF

Table 4.2: CRiB's list of atomic migration services

cial interest. The SIARD tool[14] converts relational databases into XML. Xena[15] converts files into open, publicly documented formats specifically for long-term digital preservation. Finally, the last two items in the table are accessing conversion tools that are run inside an emulated environment using Dioscuri[16], a modular emulator designed for digital preservation [vdHvW05].

CRiB

CRiB is a Service Oriented Architecture designed to assist institutions in the implementation of migration-based preservation interventions. It is publicly available[17] and described in detail in [FBR06] and [FBR07]. A query to CRiB using the PRONOM unique identifier obtained from the Format Identification Service yields a list of both atomic and chained migration services that can convert files from the input format to other more desirable preservation formats. Table 4.2 shows the 39 atomic services which are currently deployed. By composing these migration services, we get an overwhelming number of 7345 possible migration paths. For example, 147 atomic and chained migration services are available for migrating JPG images to 8 different file formats. However, it has to be noted that the majority of these services are not very useful, and only a fraction of the chained migration paths yield practical results.

CRiB offers migration services for standard object types such as images and documents. To this end, it relies on open software such as ImageMagick[18] and sam2p[19] running on Unix, but also offers Windows-based migration services for office documents. The CRiB system itself can be distributed across multiple servers running the respective platforms. The service facade provides a unified interface to these services through the tool wrappers.

[14] http://www.bar.admin.ch/dienstleistungen/00823/00825/index.html?lang=en
[15] http://sourceforge.net/projects/xena
[16] http://dioscuri.sourceforge.net/
[17] http://crib.dsi.uminho.pt
[18] http://www.imagemagick.org/
[19] http://pts.szit.bme.hu/sam2p/

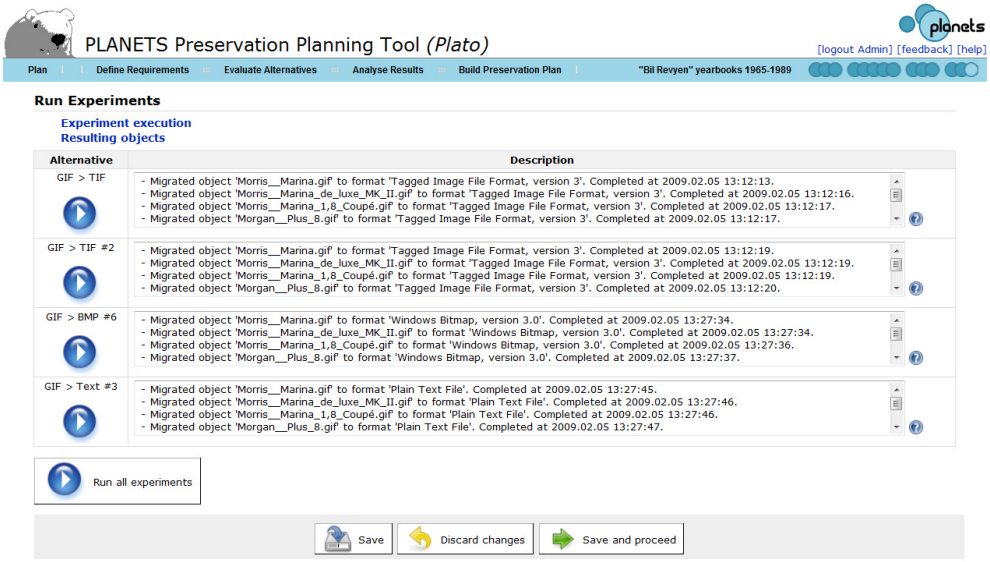


Figure 4.12: Plato showing migration service reports

MiniMEE

The existing migration services in Planets and CRiB perform very useful operations. However, they do not provide exact reports of the operations that led to the conversion results. Using the services, we obtain converted files, but we do not know some of the important characteristics of the conversion process. To overcome this issue, we have developed a controlled quality-aware migration environment called MiniMEE (Minimal Migration and Emulation Engine), which has been integrated with the planning tool. We will discuss the architecture and features of this environment in Section 6.4.

Integration

For the decision maker, the integration of the various migration engines means that the invocation of the candidate components is reduced to a *play* button that applies all components to the defined samples, as shown in Figure 4.12.

Technically, web service integration is not always straightforward. Due to the inherent incompatibilities between different frameworks and environments, web service adaptors might need to use specific implementations of the web service stack to access various services. For example, CRiB is using version 1.4 of the Apache Axis implementation of the web service stack, which uses *RPC/encoded* request transmission and does not properly support the

recommended mode *document/literal*. The framework of Planets is be based on the JBoss Application Server and thus used to rely on the JBoss-WS web service implementation, which does not fully support the older *RPC/encoded* request transmission.[20] A service adaptor is therefore needed that makes use of the Axis client to generate an *RPC/encoded* SOAP request compatible to CRiB. Similarly, the final release of the Planets framework moved to the Metro stack[21] to increase performance and scalability, which requires a different adaptor as well.

4.3.2 Emulation

Migration, i.e. file conversion, as a straightforward in/out operation lends itself naturally for being made accessible as a web service. Emulation as the second major preservation strategy is not as readily convertible. In many cases, the effort needed just to *evaluate* an emulation software is viewed as prohibitive due to the often complex setup procedure that are required even for rendering just a single file. Thus, emulation is sometimes not even considered as a potential solution, even though it might be a feasible approach. Remote access to emulation software installed on dedicated host machines thus can greatly support the process of evaluating different emulation engines for digital preservation purposes.

GRATE is a webservice written in Java/PHP/Perl and JavaScript (AJAX) and allows for location-independent remote access to designated emulation services. Figure 4.13 shows a high-level overview of the main components of the distributed emulation service infrastructure. Not shown in this diagram are the planning tool and emulation connector services contained in the component integration adaptors.

The GRATE client consists of two components, the GRATE Java applet and a Java Tight VNC client applet, embedded in PHP/JavaScript/AJAX code. Tight VNC[22] is used for remote desktop access to the emulator. Since Java applets are platform independent, every Java-enabled web-browser is suitable for running the GRATE client. This client communicates with the GRATE server component, which is responsible for session management (establishing and terminating VNC sessions, executing emulators, delivering meta-information, etc.) as well as transporting uploaded digital objects into the emulated environments. GRATE is Java-based and therefore portable. It is currently running on a Linux-Apache-MySQL-Perl-PHP (LAMPP) system. Client-server communication takes place via TCP; it is possible to input key commands into the browser, which are remotely injected into the running emulator. Injecting digital objects to be rendered is accomplished by mounting virtual drives containing the objects to be rendered. The ac-

[20] `http://labs.jboss.com/jbossws/docs/jaxws_userguide-2.0/index.html`
[21] `https://metro.dev.java.net/`
[22] `http://www.tightvnc.com/`

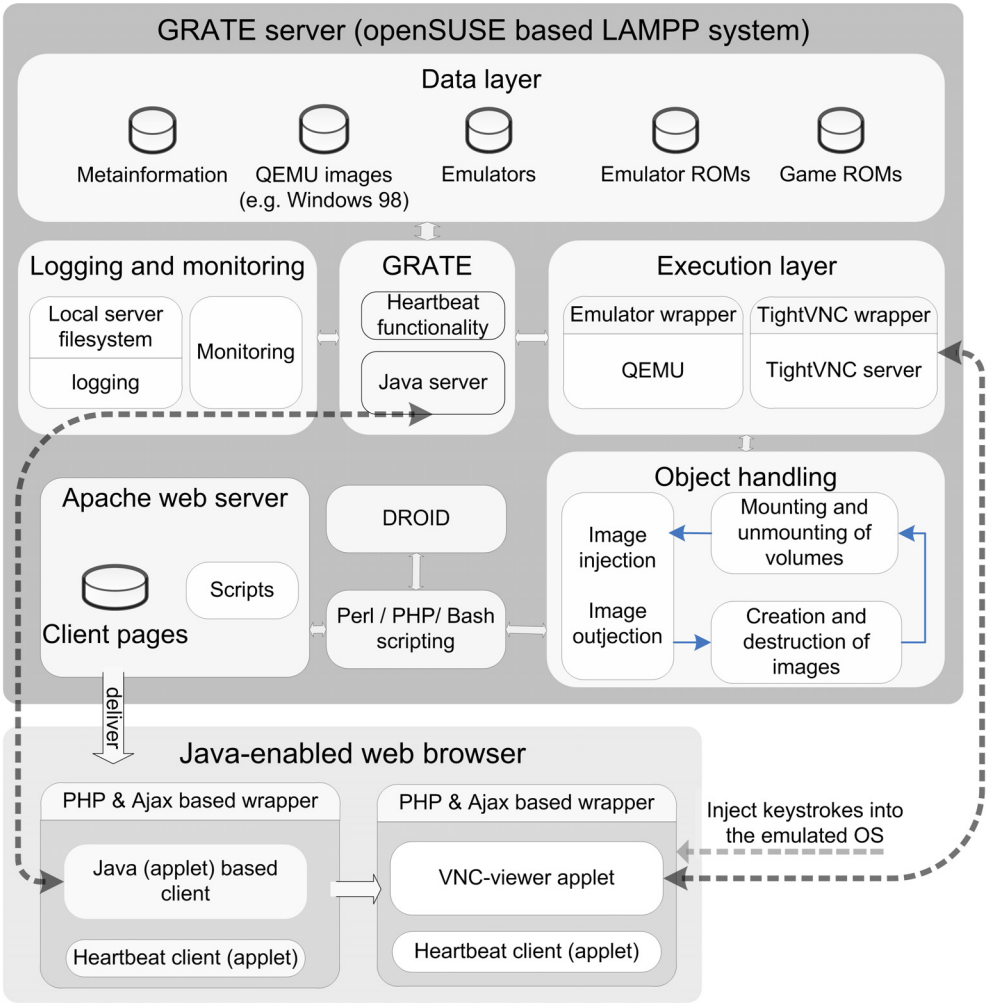


Figure 4.13: GRATE architecture [BKK+09b]

tual emulated image, e.g. of a Windows 95 installation, then contains a listener which automatically opens the encountered object. Table 4.3 gives an overview of some of the formats currently supported by the Windows 98 images running on QEMU[23].

This combination of virtual machine allocation on a pre-configured server with remote access to the emulators can reduce the total amount of time needed for evaluating a specific emulation strategy from many hours to a single click. To render a sample object within an emulated environment

[23] Note that for some formats such as RAW, only specific camera profiles are supported, while EXE and DLL means that the contained applications can be displayed. For video formats, a number of codecs are currently installed, but not listed here.

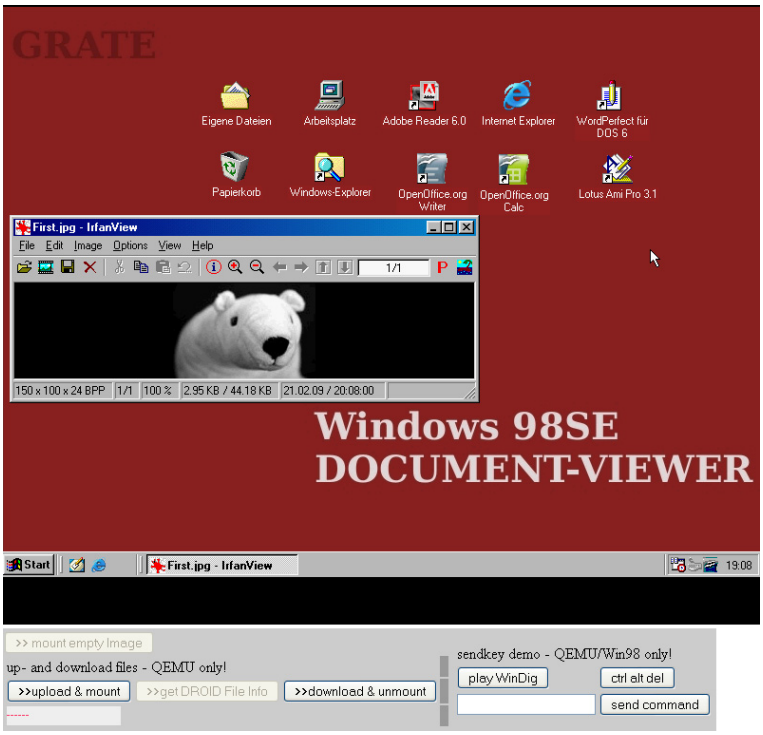


Figure 4.14: GRATE showing an injected PNG file in an image viewer

during the experiment phase of the planning procedure, the user initiates the experiment execution. The planning tool then injects the sample objects one by one into the GRATE environment, and the browser is redirected to a web server delivering the rendered screen, for example showing Figure 4.14.

4.3.3 Characterisation

While component integration for actions focuses on the actual preservation action tools doing migration and emulation, characterisation adaptors are concerned with various aspects of measurements. The most prominent examples are significant properties extractors. A number of characterisation tools for this purpose are integrated in the planning tool.

- The file format is identified by DROID, the Digital Record Object Identification tool[24].
- More detailed properties are extracted by JHOVE, the JSTOR/Harvard Object Validation Environment[25].

[24] `http://sourceforge.net/projects/droid/`
[25] `http://hul.harvard.edu/jhove/`

Video/Audio	Images	Documents
AIF, AU, SND, MED, MID, MP3, OGG, RA, WAV, ASF, AVI, MOV, MP4, MPG, MPEG, WMA, WMV	ANI, CUR, AWD, B3D, BMP, DIB, CAM, CLP, CPT, CRW/CR2, DCM/ACR/IMA, DCX, DDS, DJVU, IW44, DXF, DWG, HPGL, CGM, SVG, ECW, EMF, EPS, PS, PDF, EXR, FITS, FPX, FSH, G3, GIF, HDR, HDP, WDP, ICL, EXE, DLL, ICO, ICS, IFF, LBM, IMG, JP2, JPC, J2K, JPG, JPEG, JPM, KDC, LDF, LWF, Mac PICT, QTIF, MP4, MNG, JNG, MRC, MrSID, SID, DNG, EEF, NEF, MRW, ORF, RAF, DCR, SRF/ARW, PEF, X3F, NLM, NOL, NGG, PBM, PCD, PCX, PDF, PGM, PIC, PNG, PPM, PSD, PSP, PVR, RAS, SUN, RAW, YUV, RLE, SFF, SFW, SGI, RGB, SIF, SWF, FLV, TGA, TIF, TIFF, TTF, TXT, VTF, WAD, WAL, WBMP, WMF, WSQ, XBM, XPM	PDF, ODT, OTT, SXW, STW, DOC, DOCX, DOT, TXT, HTML, HTM, LWP, WPD, RTF, FODT, ODS, OTS, SXC, STC, XLS, XLW, XLT, CSV, ODP, OTP, SXI, STI, PPT, PPS, POT, SXD, ODG, OTG, SXD, STD, SGV

Table 4.3: Formats supported by the Windows images deployed in GRATE

- Both of these tools are also wrapped by FITS, the File Information Tool Set[26]. FITS includes several other metadata extractors such as the ExifTool[27], the National Library of New Zealand Metadata Extractor[28], and FFident, a Java metadata extraction / file format identification library[29]. FITS partially normalises the output of these tools by applying a set of extensible XML transformation rules.

Apart from these tools, other aspects of interest include performance measurements of tools and access to shared registries containing trusted information and accumulated experiece. Since these measurement tools are of central importance and considerable complexity, Chapter 6 will explore them in detail.

4.4 Deployment

Figure 4.15 shows a possible deployment of the distributed infrastructure. The shown deployment consists of seven server instances; additional registries and services can be dynamically added and registered in the planning tool. The Planets server instance on the top-left side corresponds to the application server running the main deployment of Plato[30]. The interoperability framework provides features such as a workflow execution engine, a data registry based on a Java Content Repository (JCR)[31] implementation, and services such as user management, Single-Sign-On, persistence, and logging.

The characterisation tools mentioned in Section 4.2.1 are invoked directly on the server, while the XCL tool suite in this example resides on a separate server running another Planets instance. This server also contains a number

[26] http://code.google.com/p/fits/

[27] http://www.sno.phy.queensu.ca/~phil/exiftool/

[28] http://meta-extractor.sourceforge.net/

[29] http://web.archive.org/web/20061106114156/http://schmidt.devlib.org/ffident/index.html

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

[31] http://jcp.org/aboutJava/communityprocess/final/jsr170/index.html

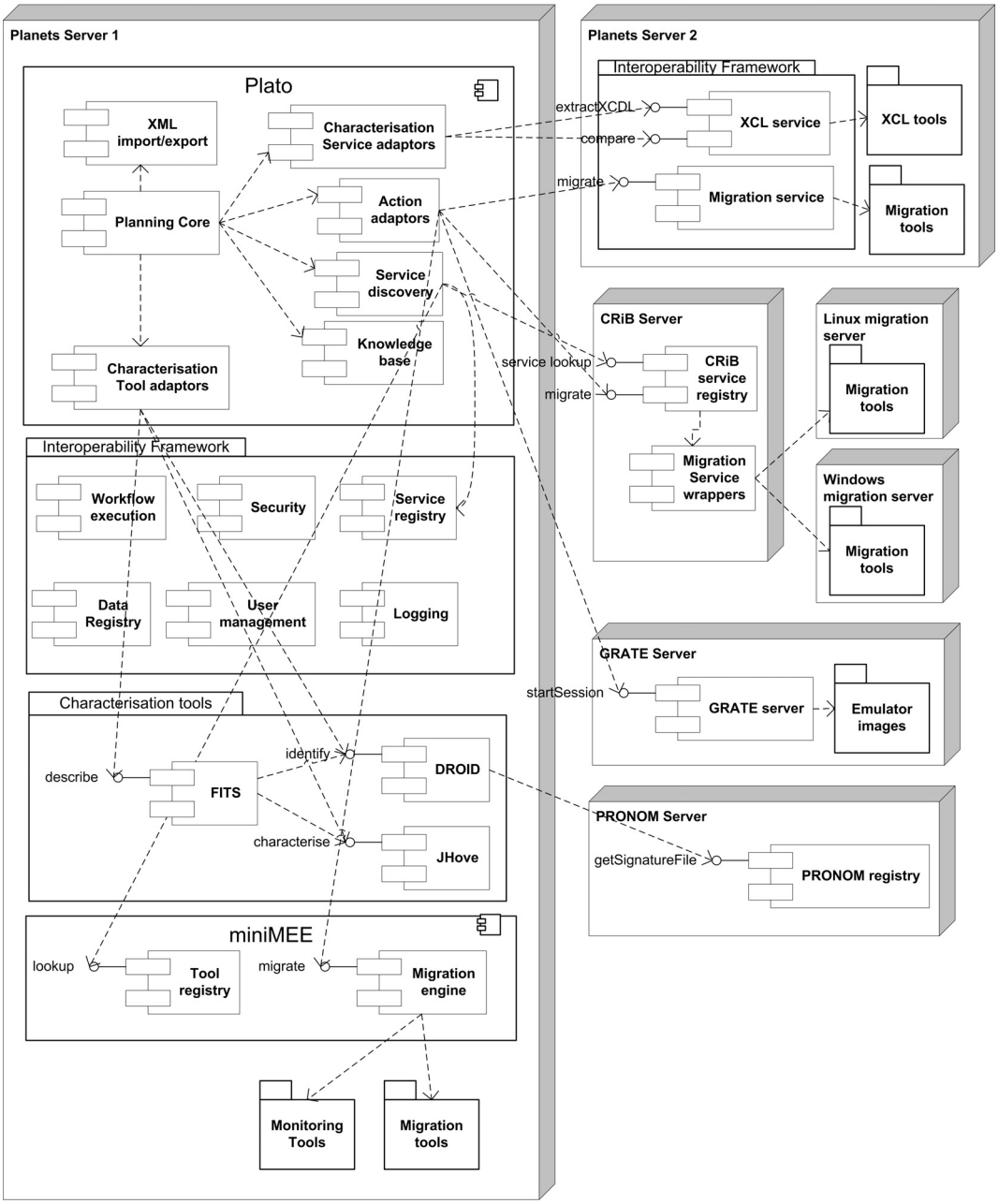


Figure 4.15: A distributed preservation planning deployment

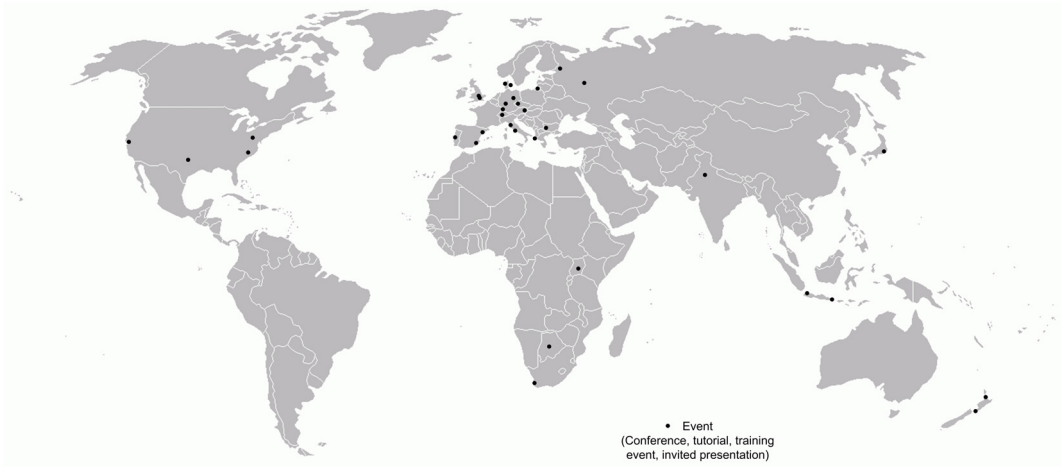


Figure 4.16: Event venues where Plato was presented

of migration tools accessible through a web service interface. The main Plato server is furthermore running a MiniMEE engine containing quality-aware migration tools executed in a controlled environment.

The example further illustrates several additional servers: CRiB and and GRATE are running on dedicated environments, and the identification tool DROID is accessing the publicly available PRONOM server for updating the signature file which describes patterns for file format identification.

4.5 Summary and Takeup

This chapter has provided a practical introduction of the web-based planning tool Plato, which implements the method described in Chapter 3. We gave an overview of the main features and described the architecture for integrating action and characterisation components. The code of the planning tool is available under an open license and can be obtained from the website[32].

Since its first public deployment in Spring 2008, the planning tool has received considerable attention in the digital preservation community. Plato shared the *Best Demo* award at ECDL 2008[33] with the search engine Summa[34] and was presented at a number of venues, including

- Digital library conferences in the computing field, such as JCDL, ECDL, ICADL, ICDL, and RCDL;

[32] `http://www.ifs.tuwien.ac.at/dp/plato/intro.html`
[33] `http://www.ecdl2008.org/`
[34] `http://sourceforge.net/projects/summa/`

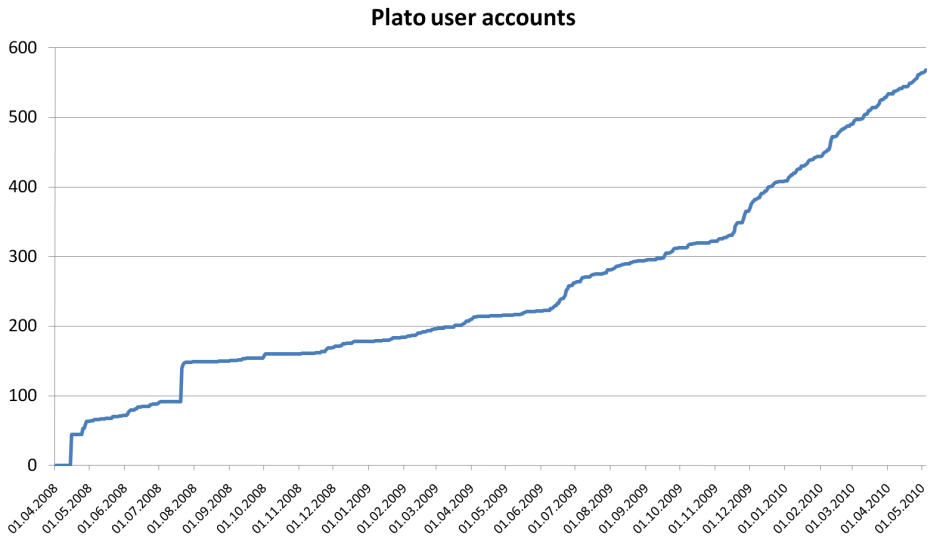


Figure 4.17: Number of user accounts between April 2008 and April 2010

- Digital preservation workshops and training events conducted in Europe by organisations such as DPC, nestor, Planets, and wePreserve;
- IT venues such as CeBIT, IST Africa, and SUN PASIG meetings; and
- Digital preservation and digital curation conferences and workshops, such as iPRES and DigCCurr, in Europe and the US.

Figure 4.16 shows a map of event venues where the planning tool was presented. An up-to-date list of events is available at the Plato homepage[35]. The development of user accounts since the public reference deployment was published is shown in Figure 4.17.[36] In June 2009, the planning tool had 241 users who had created 135 preservation planning projects. By January 2010, there were over 430 users from Europe, North America, Russia, Asia, and Oceania, and the number of plans had almost doubled. Currently, there are over 560 user accounts registered from 45 different top level domains, and the entry page consistently comes up number 1 on a google query "preservation planning". Figure 4.18 shows the distribution of user accounts. The strongest groups are from the UK, Germany, Austria, The Netherlands, and US-American universities (`.edu`).

[35] `http://www.ifs.tuwien.ac.at/dp/plato/intro_events.html`

[36] Note that the two sudden surges in 2008 come from planning workshops where user accounts were created for all participants. From November 2008 on, all accounts were created by the users themselves using a self-registration feature.

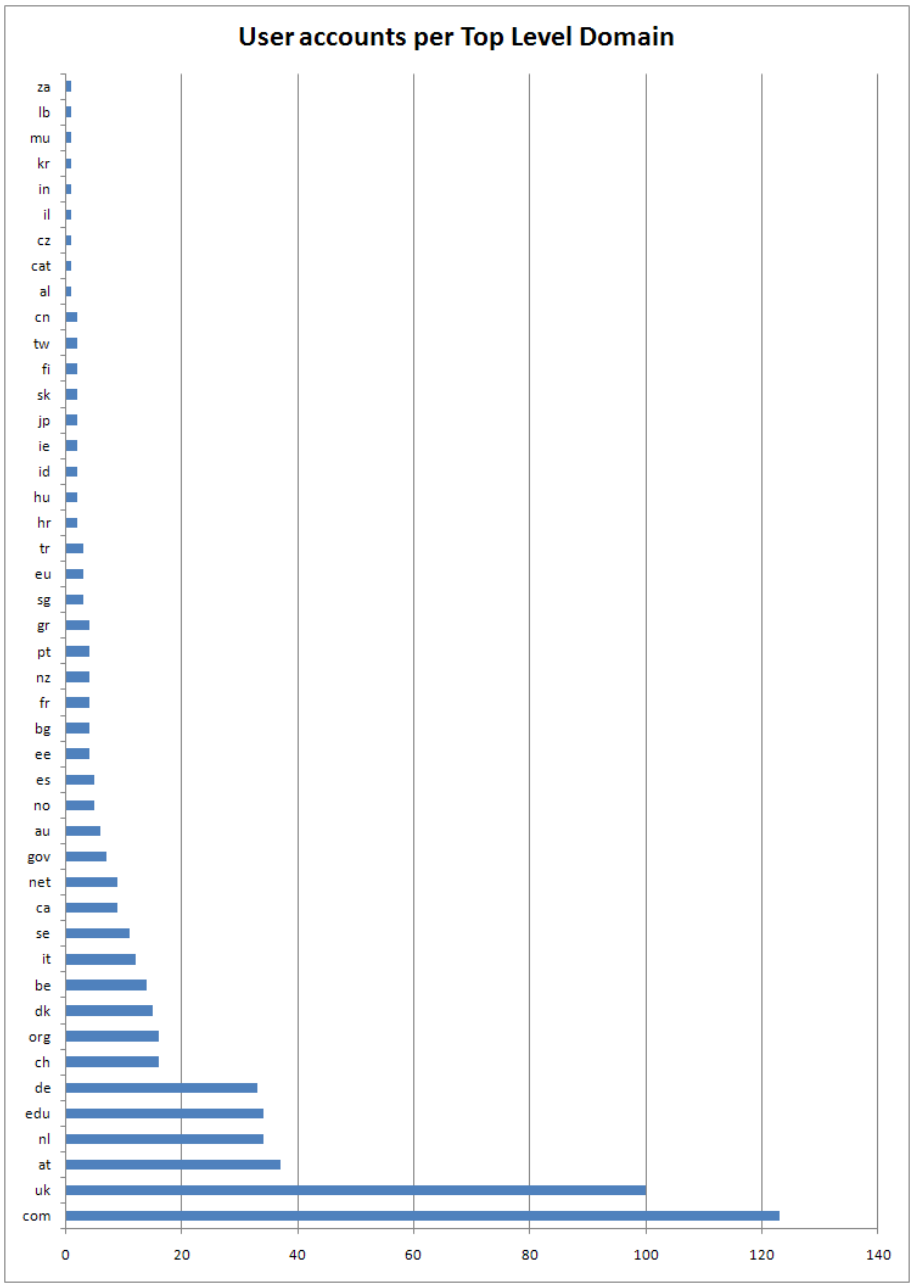


Figure 4.18: Distribution of user accounts according to top level domains

Chapter 5

Case studies

In the last chapter, we have provided a practical overview of the planning tool and noted that a significant number of users have started using the tool. This chapter provides a few examples of applying the planning method and tool to real-world cases and discusses lessons learned.

In Section 5.1 we first discuss a case study in defining significant properties for a particularly volatile set of objects: interactive electronic art. We contrast this in Section 5.2 to a subsequent case where we evaluated preservation strategies for console video games and compared emulation approaches to migration. Section 5.3 presents an evaluation of relational database preservation strategies, focusing on the complex definition of database properties to be preserved. Finally, Section 5.4 presents four exemplary recent case studies that created preservation plans for images in four different instutions, and discusses why the evaluation of the scenarios arrived at three different recommendations. Section 5.5 discusses the lessons learned from these case studies and describes common misperceptions and typical mistakes in applying the approach. Section 5.6 conducts a critical assessment of current shortcomings and gaps, and hence sets out directions for further work in this thesis and beyond.

Note that while most studies were evaluating components without a business-driven case of urgent action needs, three of the image preservation case studies were delivering productive business decisions.

5.1 Significant properties of interactive electronic art

5.1.1 Introduction

Traditional memory institutions primarily own collections of digitised material from analog sources and large homogeneous collections of electronic documents in widely adopted and well-understood file formats. In contrast,

collections of born-digital art pose a whole new problem field. Electronic art is extremely complex to preserve: The employed media as well as the file formats are heterogeneous, complex, and often proprietary and unstable. Artists cannot be obliged to conform to submission policies that prescribe formats and standards, which yields highly heterogeneous collections of proprietary file formats.

This section presents findings of a pilot project dealing with the preservation of born-digital multimedia art. Specifically, we focus on a collection of interactive artworks held by the Ars Electronica[1]. We describe the context of the collection and the specific challenges that interactive multimedia art poses to digital preservation. We then focus on the requirements on significant properties that potential preservation strategies have to fulfil in order to be fit for purpose in the given setting.

The challenge of preserving born-digital multimedia art, which is inherently interactive, virtual, and temporary, has been an actively discussed topic in the last decade. Besser reports on the longevity of electronic art in [Bes01]. In 2004, the ERPANET project organised a workshop [ERP04] on archiving and preservation of born-digital art. The Variable Media Network, a joint effort founded by institutions such as the Guggenheim Museum New York and the Berkeley Art Museum/Pacific Film Archives, investigated properties of an artwork that are subject to change and developed tools, methods and standards to implement new preservation strategies for unstable and mixed media [DIJ04]. The most prominent results of this initiative is the Variable Media Questionnaire [Var10], developed at the Guggenheim Musem New York, which assists artists and curators in understanding which properties of an artwork are subject to change and how these should be handled in the best possible way. Guggenheim and the Pacific Film Archive participated in the project "Archiving the Avantgarde"[bam07], developing ways to catalog and preserve collections of variable media art.

In the field of computer science, the most notable work has been carried out in the PANIC project [HC06, HC04] which developed preservation strategies for multimedia objects [HC03]. However, they focus on dealing with composite objects that contain different content; interaction is not covered. Yeung et. al. discusses challenges and solution approaches to preserving digital art [YCG08]. A recent state-of-the-art report[MKB09] conducted by the Planets project is summarised in [MK10].

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 conflict between the transformation necessary to keep the work accessible, and the desired authenticity of each piece of art [Dep01]. Jones [Jon04] reports on a case study which used hardware emulation to recreate one of the first interactive video artworks.

[1] `http://www.aec.at`

Emulation is often able to retain the original appearance of the digital object, and some claim it is the ideal preservation solution [Rot99]. The main points of criticism are its complexity and the fact that intellectual property rights might prevent the creation of emulators [Bea99, Gra00].

The main obstacle to the second prominent approach, migration, in this context is the diversity and complexity of obsolete file formats that are used in the field of digital art. Depocas [Dep01] argues that efforts to preserve born-digital media art always have to be based on structured documentation and adds that often the documentation is the only thing that remains.

5.1.2 A Real-World Case: Ars Electronica

More and more modern museums hold pieces of born-digital art. The Ars Electronica in Linz, Austria, is one of the most prominent institutions in the field of electronic art. It owns one of the world's most extensive archives of digital media art collected over the last 25 years.

At the time of conducting this case study in 2007, the collection of the Ars Electronica contained more than 30.000 works and documentation videos and was growing at a rate of over 3000 pieces per year. Of these works, about 6200 pieces had been deposited as CDs containing multimedia and interactive art in different formats like long-obsolete presentation file formats with interactive visuals, audio and video content. The CDs are divided into the categories Digital Music (4000), Computer Animation (1000), and Interactive Art (1200). These collections pose extreme problems to digital preservation due to their specific and complex characteristics. The main issues arising in this context are the following.

1. The collections are highly heterogeneous, there is no common file format. Instead, digital art ranges from standard image and video files to specifically designed, proprietary software pieces which are sometimes highly dependent on a specific environment. Of each of the many formats, there is only a few examples, making automation and large-scale application a real challenge.

2. Many of the artworks are integrated applications, for which the underlying file format can not be easily identified. For example, some interactive artworks combine multimedia content with viewer applications specifically designed to render the contained objects. Other pieces even deal with the issues of digital deterioration, damaged content and the like.

3. 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.

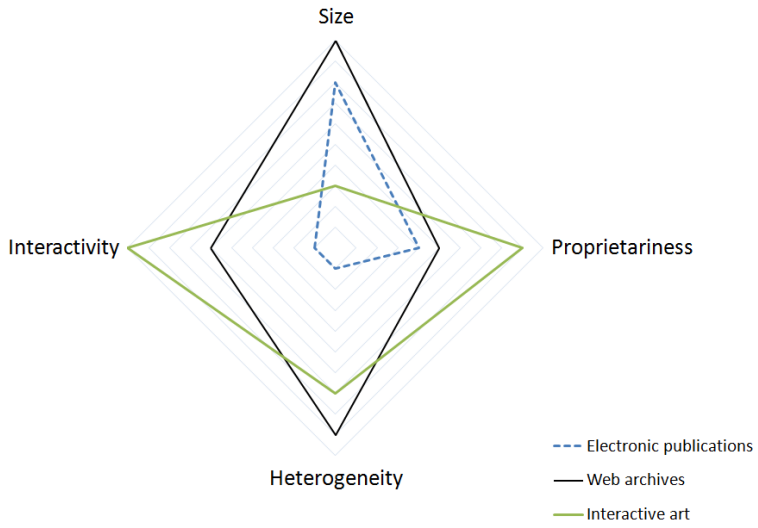


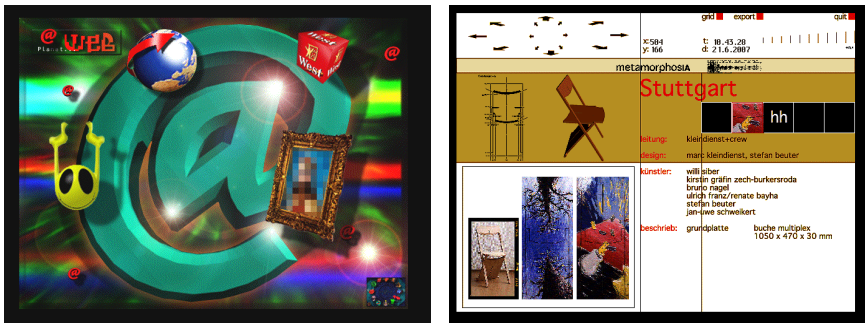
Figure 5.1: Comparison of content characteristics

Contrary to traditional digital preservation endeavours, these specific issues also bring about the need for particular actions. For certain objects, it may be necessary to involve creators in the definition of requirements to ensure that their intentions are communicated correctly. The challenge is to keep the authentic message of their artwork, while potentially transforming the representation of the piece of art and its presentation. Some works might only be preservable by developing custom software particularly for this purpose.

Figure 5.1 illustrates distinct characteristics of some typical content types, setting digital art in relation to more common objects. Collections of electronic publications are fairly large and the contained formats can be quite complex; but the collections are generally homogeneous and tend to be in standardised formats. Interactivity is not a major concern, and there are a number of migration tools available. In contrast, web archives are extremely heterogeneuous and potentially huge. Collections of interactive art are very small in comparison, but the combination of proprietaryness, heterogeneity and interactivity means that providing access is a challenging problem.

In a joint effort with the Ludwig Boltzmann Institute Media.Art.Research[2], we investigated possible approaches to deal with the preservation of born-digital interactive art. The aim was to not only preserve these pieces of art over the long term, but also to make them accessible in a satisfying form on the web. In a pilot study, we concentrated on a sub-collection of the large collection the Ars Electronica owns. This collection contains about

[2] `http://media.lbg.ac.at/en`



(a) Kolb: Cycosmos (b) Kleindienst,Beuter: Metamorphosia

Figure 5.2: Sample interactive artworks from 1997 [BKKR07]

90 interactive presentations in the formats Asymetrix Compel, Asymetrix Toolbook, and Macromedia Director. The companies that created these formats had ceased to exist; only the latter format was supported by current software [Ado10].

The aim of the pilot project was to find means of preserving the original intention of the artists as well as the user experience and thus truly preserve the original artwork. To achieve this, we applied the preservation planning approach described in Chapter 3 to analyse the requirements on preserving interactive art. In a series of workshops with curators, art historians, computer scientists, preservation specialists, and management, the first phase of the planning process was completed. Figure 5.2 shows screenshots of two exemplary sample records that were chosen as part of this process. These sample objects are used for identifying requirements and evaluating the performance of different preservation strategies.

Figure 5.3 provides an overview of the essential object characteristics that were identified, and also documents the weights that have been assigned to the upper levels of the tree hierarchy. Object characteristics are divided into the aspects content, appearance, structure, behaviour, and context. Naturally, the primary focus lies on the content of the artworks, such as the contained text, images, and sounds. The second most important criterion is the completeness of the navigational structure that forms the backbone of each interactive artwork. A purely linear recording of an interactive piece of art will most probably not capture the true spirit and the spectator's experience. This interactivity is also by far the most important criterion when it comes to behavioural characteristics.

Figure 5.4 details some aspects of the object characteristics as they are displayed in the planning tool and provides measurement units to illustrate the quantitative nature of the evaluation process.

A particularly important aspect is the measurement of interaction fea-

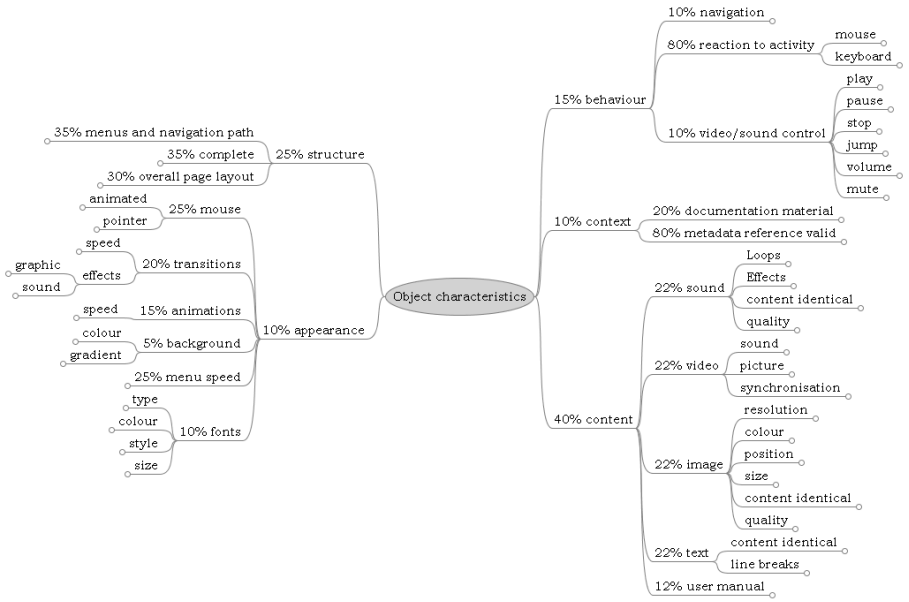


Figure 5.3: High-level weighted object characteristics for interactive art

tures and the degree to which they have been preserved by a preservation action. In principle, interactive presentations exhibit two facets: They have a graph-like navigation structure, and they allow the user to navigate along the paths. Different strategies of preserving an interactive presentation will show different strengths and weaknesses in preserving these characteristics. For example, migrating an interactive presentation to a collection of images and videos and documenting the navigational structure externally will preserve the complete structure and the possibility to navigate along the paths, but miss the interactivity. The structural aspect of this is measured in the criterion *menus and navigation path*. The interaction is covered by the behavioural criterion *navigation*, which can take one of the values *interactive and integrated, navigable*, or *none*.

First research on available solutions revealed that no off-the-shelf solution will be applicable for all objects. We are investigating the development of a pathfinder application that acts as a robot to control the mouse and other inputs, while recording the screen and comparing screenshots to detect changes. Combining depth-first and breadth-first graph construction, this may be able to migrate interactive content with the described characteristics successfully to rich HTML or the Synchronized Multimedia Integration Language[3] (SMIL). Another approach is the usage of emulation, potentially combined with access architectures such as GRATE.

[3] `http://www.w3.org/AudioVideo/`

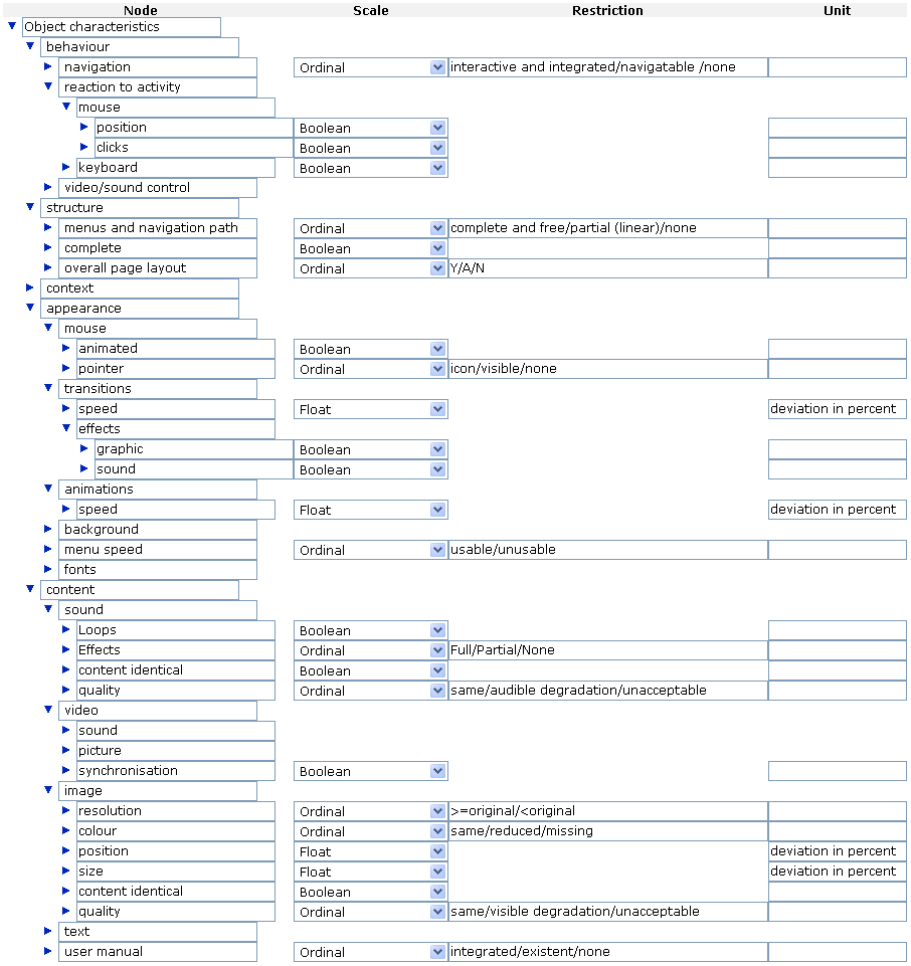


Figure 5.4: Selected object characteristics for interactive multimedia art

5.2 Interactive games

Building on this work, a related case study was carried out in the field of console video game preservation. It analysed the challenges of proprietary hardware and unavailable documentation as well as the considerable variety of media and non-standard controllers, and used the planning approach to evaluate emulation and migration of console video games using one objective tree. Experiments were carried out to compare different emulators as well as other approaches. We carried out comparison of potential actions for a single console video game system, across different console systems of the same era, and finally across systems of all eras. A detailed discussion of this case study is presented in [GBR08]. The top levels of the requirements tree are shown in Figure 5.5.

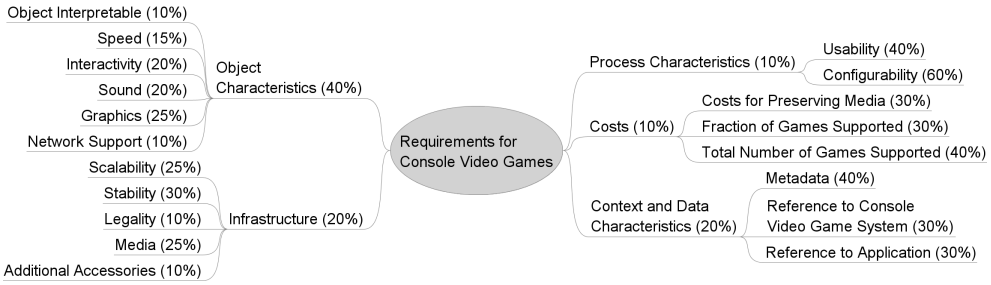


Figure 5.5: Top level requirements for preserving console video games

The selection of samples can be particularly critical with interactive content and emulation, because there is substantial variation in the principal support and quality of different games across the emulators. It is quite common for an emulator to only work well with a fraction of the games. This is thus also an important cost factor, as shown in Figure 5.5.

The analysis of available actions compared a number of emulators and contrasted their strengths and weaknesses with a simple conversion to a video of the game play. The results showed that the preservation planning approach can be used to evaluate both emulation and migration in the same planning process and compare different strategies within one objective tree. Evaluation revealed that emulation works in principle well for early console video games, but needs to overcome several problems in order to be generally usable as a digital preservation action in this context: Emulators generally have no built-in metadata support, are not platform-independent (and thus themselves present preservation problems on their own), and do not work well for emulating newer games. Stability and metadata handling need to be improved.

Table 5.1 shows the overall evaluation results. Since the behaviour of emulators varies considerably with different games, we include results for all sample objects. We see that MESS is eliminated in Weighted Multiplication (WM) because it fails to render the sample "Starfox" properly. The video migration (VLC/MP4) is eliminated due to the loss of interactivity. ZSNES and SNESX have very similar scores for each sample object and thus also are very close in the overall score. A detailed discussion of the tree and evaluation results can be found in [GBR08].

5.3 Database preservation

This case study was carried out with a national archive in Europe. We analysed the requirements for preserving relational databases in MS Access format and compared the action of migrating a set of relational databases

Alternative	Sample object	WSSample	WMSample	WSTotal	WMTotal
ZSNES 1.51	Super Mario World	3,45	2,75	3,28	2,68
	Super Scope 6	3,30	2,70		
	Starfox	3,38	2,78		
SNES9X 1.51	Super Mario World	3,43	2,82	3,31	2,70
	Super Scope 6	3,28	2,68		
	Starfox	3,38	2,78		
MESS 0.119	Super Mario World	3,56	2,88	2,68	0,00
	Super Scope 6	3,47	2,79		
	Starfox	2,47	0,00		
VLC 0.8.6c/MP4	Super Mario World	4,65	0,00	4,65	0,00

Table 5.1: Evaluation results for preserving games for the Nintendo SNES

to an openly specified XML format with the alternatives of exporting the content to Comma Separated Values (CSV) and leaving the databases unchanged. The objective tree contains extensive specification of desired contextual properties of the databases, as shown in Figure 5.6. In this archival scenario, descriptive metadata are of course seen as essential elements. But other aspects such as the archival process itself are considered important as well, and to ensure future understandability, a documentation of the data model and the data dictionary must be preserved.

The actual content characteristics are described in Figure 5.7. The hierarchical structure is organised along the encountered data types and includes typical structural database elements such as views, users, roles, and contraints.

Some of the scales used for evaluation are of particular interest. For example, user defined datatypes may be either preserved completely, converted to standard datatypes, or lost; and views may be converted to SQL99, left unchanged, converted to a table, or lost. It depends on the organisational context whether conversion to standard SQL is preferred to conversion to a table, which resolves the dependencies at the cost of redundant storage. In our case, the preference was conversion to SQL99 with a utility of 5, followed by leaving the view declaration unchanged (4) and conversion to a table (3), while the loss of views was unacceptable (0).

Figure 5.8 shows further evaluation and transformation details for the format characteristics. The left side presents the evaluation values of each alternative; the middle column *Transformer* contains the utility function mapping each possible value to a utility; and the column to the right shows the transformed utility values for each alternative. All alternatives show some weaknesses. Archival to XML gets a low utility score for behaviour, while CSV export is weak on aspects of encoding and format standardisation. The latter is also a severe drawback of leaving the databases in their original format.

The top level results of the evaluation are shown in Figure 5.9. We see that CSV export is unacceptable due to its performance in the *content* branch. The primary reason lies in the loss of columns that contain large binary data (BLOBs). CSV also shows significant weaknesses in terms of

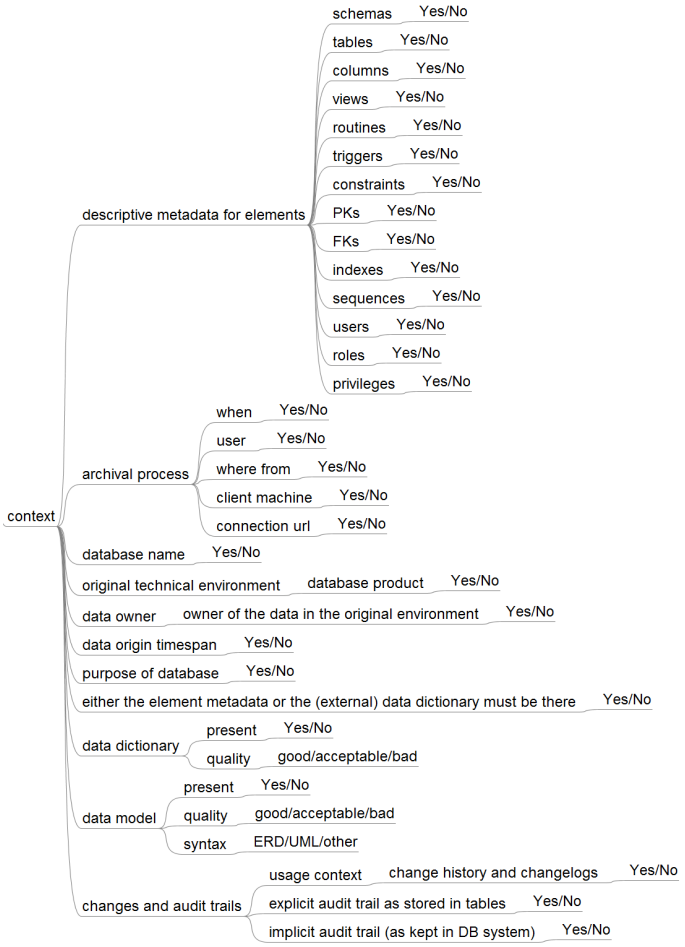


Figure 5.6: Context requirements for relational databases

schema definitions, both in terms of structure and datatypes, and of course does not support views and other important aspects.

The benefit of archiving to a standardised XML schema in that case is seen in the format characteristics. Plain text readability as a safe fall-back strategy, as well as standardisation and open availability of the format are the primary factors that contribute to an advantage of XML archival compared to the original format. This is considered more important than the small advantage that the original database shows in terms of content.

It has to be noted that while the XML archiving solution does not preserve behaviour, this is hardly considered relevant in our specific scenario, where the focus is purely on preserving the factual content of the original databases together with a documentation of the original access mode. The weighting of criteria reflects this: Content and context account for 84% of the

Figure 5.7: Content requirements for relational databases

weights within the object characteristics and 53% of the overall result in the tree. This is strikingly different from other scenarios where behaviour is an essential element of the objects, such as interactive art and games discussed above. For example, in interactive art, content and context accounted for only 50% of the significant properties of the objects. These differences in weightings reflect the specific environments and usage scenarios in question.

5.4 Preservation plans for images: Four cases, three solutions

This section discusses four related exemplary case studies, each seeking the optimal preservation solution for large collections of scanned images. These case studies took place in four different national libraries in Europe.

Significant properties of images are relatively straightforward to define

Object characteristics > behaviour

Results

Alternatives	1
Archive to XML	not preserved
Keep original DB	preserved
CSV export	not preserved

Transformer

Ordinal Value	Target Value
preserved	-> 5.0
not preserved	-> 1.0

Transformed Results

Alternatives	1	Aggregated	Co
Archive to XML	1	1	
Keep original DB	5	5	
CSV export	1	1	

Aggregation mode: Worst result

Format characteristics > datatype support for column types: string, number, datetime

Results

Alternatives	1
Archive to XML	Yes
Keep original DB	Yes
CSV export	No

Transformer

Ordinal Value	Target Value
Yes	-> 5.0
No	-> 1.0

Transformed Results

Alternatives	1	Aggregated	Comments
Archive to XML	5	5	
Keep original DB	5	5	
CSV export	1	1	

Aggregation mode: Worst result

Format characteristics > character encoding

Results

Alternatives	1
Archive to XML	unicode
Keep original DB	original encoding
CSV export	original encoding

Transformer

Ordinal Value	Target Value
unicode	-> 5.0
original encoding	-> 4.0
other	-> 3.0

Transformed Results

Alternatives	1	Aggregated
Archive to XML	5	5
Keep original DB	4	4
CSV export	4	4

Aggregation mode: Worst result

Format characteristics > time encoding

Results

Alternatives	1
Archive to XML	ISO8601
Keep original DB	original encoding
CSV export	other

Transformer

Ordinal Value	Target Value
ISO8601	-> 5.0
original encoding	-> 4.0
other	-> 3.0

Transformed Results

Alternatives	1	Aggregated
Archive to XML	5	5
Keep original DB	4	4
CSV export	3	3

Aggregation mode: Worst result

Format characteristics > readable in plain text

Results

Alternatives	1
Archive to XML	Yes
Keep original DB	No
CSV export	No

Transformer

Ordinal Value	Target Value
Yes	-> 5.0
No	-> 2.0

Transformed Results

Alternatives	1	Aggregated	Comments
Archive to XML	5	5	
Keep original DB	2	2	
CSV export	2	2	

Aggregation mode: Worst result

Format characteristics > published

Results

Alternatives	1
Archive to XML	Yes
Keep original DB	No
CSV export	No

Transformer

Ordinal Value	Target Value
Yes	-> 5.0
No	-> 1.0

Transformed Results

Alternatives	1	Aggregated	Comments
Archive to XML	5	5	
Keep original DB	1	1	
CSV export	1	1	

Aggregation mode: Worst result

Figure 5.8: Example evaluation and transformation for database archival formats

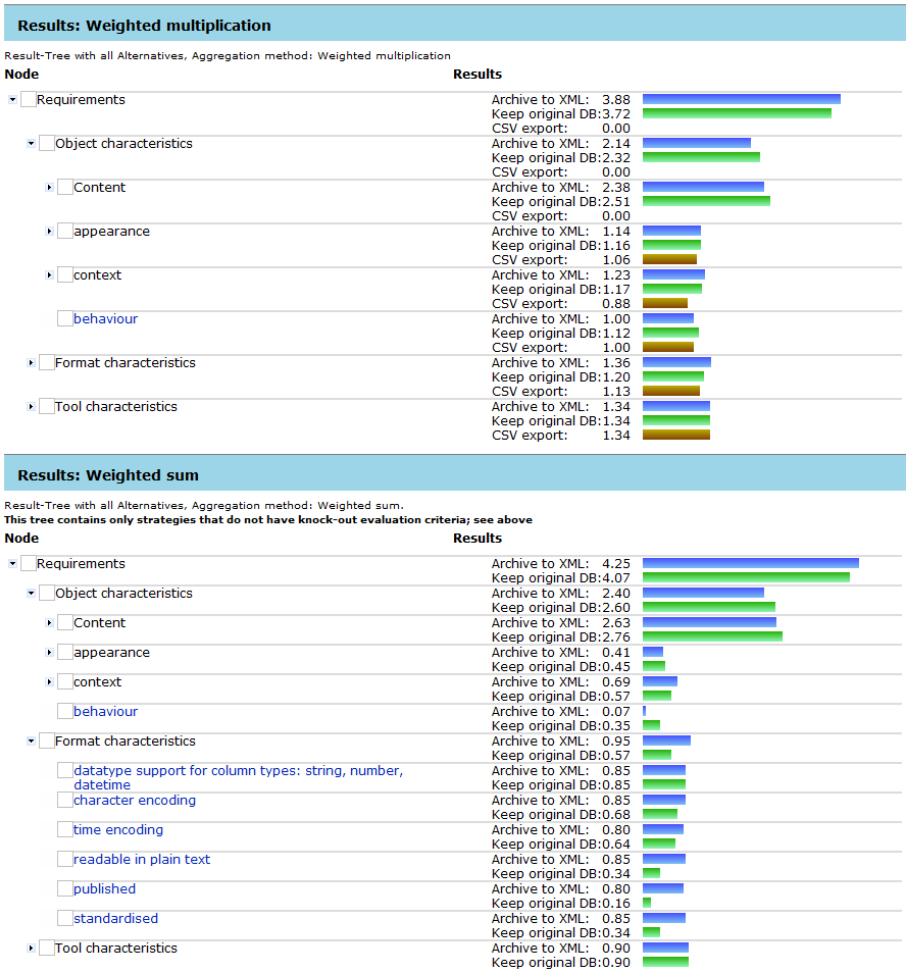


Figure 5.9: Top level results for database preservation

compared to complex objects of the types discussed in the last studies. We will thus focus our attention on the process aspects and the peculiarities that differentiate the case studies.

5.4.1 Scanned newspapers

The first case study was carried out with the British Library[4] and focused on a collection of 2 million images in TIFF-5 format with a size of about 40MB per image. The images were scanned from old newspaper pages; with 80TB of data volume this was the largest study in terms of size. Figure 5.10

[4] `http://www.bl.uk`


The Leeds Mercury.
PRINTED AND PUBLISHED BY EDWARD BAINES, AT HIS OFFICE, IN BRIGGATE, LEEDS.
SATURDAY, FEBRUARY 7, 1807.
[PRICE SIX-PENCE.

Figure 5.10: Scanned newspaper page

shows a sample image of a scanned page.[5]

Concerns were raised about the suitability of the Linear Tape Open (LTO) media on which the content was held, and the images were transferred to hard disk storage and reviewed. This move highlighted difficulties in accessing some of the tapes, and a decision was taken to transfer the material into the main digital library system. Before the ingest, it was decided to review the format of the master files to see if the current format was the most suitable or whether a migration should be performed as part of the media replacement.

Some of the high-level policies that affect the decision making in terms of file formats include

[5]Copyright held by The British Library.

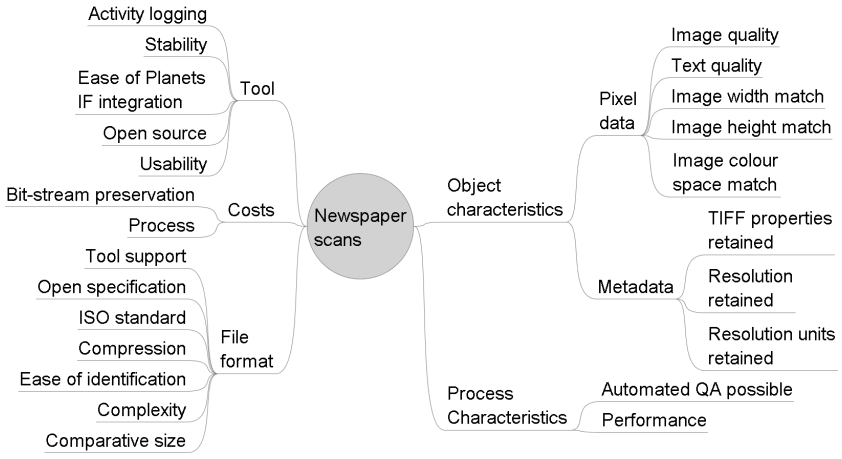


Figure 5.11: Scanned newspaper requirements tree

1. Open target formats are highly preferred,
2. Compression must be lossless, and
3. Original copies may be deleted.

The requirements tree as shown in Figure 5.11 is quite compact, as significant properties of images are not overly complex. A variety of options, including not changing the format of the images, were evaluated in a series of controlled experiments. The costs were calculated using the LIFE models[6]. Table 5.2 shows the evaluated preservation actions and their aggregated scores. Conversion to BMP was ruled out prior to the experiment phase due to expected large file sizes and lack of compression, while GIF was discarded because of the palette size limitations.

The results show that migration to uncompressed JPEG2000 (JP2) achieves a slightly higher root score than leaving the master files untouched. The reasons are that the long-term storage costs and the fact that JP2 is a recognised ISO standard [ISO04c] outweigh the process costs of converting the images. Conversion to JPEG or to compressed JP2 is violating the abovementioned policy that compression must be lossless, as included in the requirements tree under *File format – Compression*. Thus the corresponding alternatives have a multiplication score of 0.0 and are discarded as unacceptable alternatives.

[6] `http://www.life.ac.uk`

Candidate action	Weighted multiplication	Weighted sum
Leave in TIFF-5	3.01	3.46
Convert TIFF to PNG (ImageMagick)	2.72	3.27
Convert TIFF to BMP (ImageMagick)	-	-
Convert TIFF to GIF (ImageMagick)	-	-
Convert TIFF to JPEG (ImageMagick)	0.00	-
Convert TIFF to JP2 (ImageMagick)	3.44	3.69
Convert TIFF to JP2 95 (ImageMagick)	0.00	-
Convert TIFF to JP2 90 (ImageMagick)	0.00	-
Convert TIFF to JP2 80 (ImageMagick)	0.00	-

Table 5.2: Evaluation results for preservation actions on newspaper scans

Candidate action	Weighted multiplication	Weighted sum
Keep status quo (TIFF-6)	4.50	4.70
Convert TIFF to JP2 (ImageMagick)	3.71	4.09
Convert TIFF to JP2 (GraphicsMagick)	0.00	-
Convert TIFF to JP2 (Kakadu)	3.68	4.06
Convert TIFF to JP2 (GeoJasper)	3.65	4.03

Table 5.3: Evaluation results for preservation actions on scanned books

5.4.2 Scanned books

A smilar study which examined the options for preserving a large collection of images scanned from 16th-century books held by the Bavarian State Library[7] is presented in detail in [KRB+09]. The collection contains 21.000 prints with about 3 million pages in TIFF-6, totalling 72TB in size. The requirements elicitation procedure involved stakeholders ranging from the head of digital library and digitisation services to digitisation experts, library employees, and employees from the supercomputing centre responsible for the storage. The resulting requirements tree is shown in Figure 5.12. The considered actions were migration to JP2 with various conversion tools and leaving the objects unchanged. Storage itself does not pose significant constraints on this specific collection at the moment. The costs of the migration process, however, are dependent on the cost model of the computing facility to which the storage is outsourced. There, the pay-per-volume cost model depends on the volume of data that is retrieved from or (re-)ingested into the archive.

The evaluation results displayed in Table 5.3 show that leaving the images in TIFF-6 is the preferred option. This is despite JP2 having advantages such as reduced storage requirements and streaming support. The

[7] http://www.bsb-muenchen.de/index.php?L=3

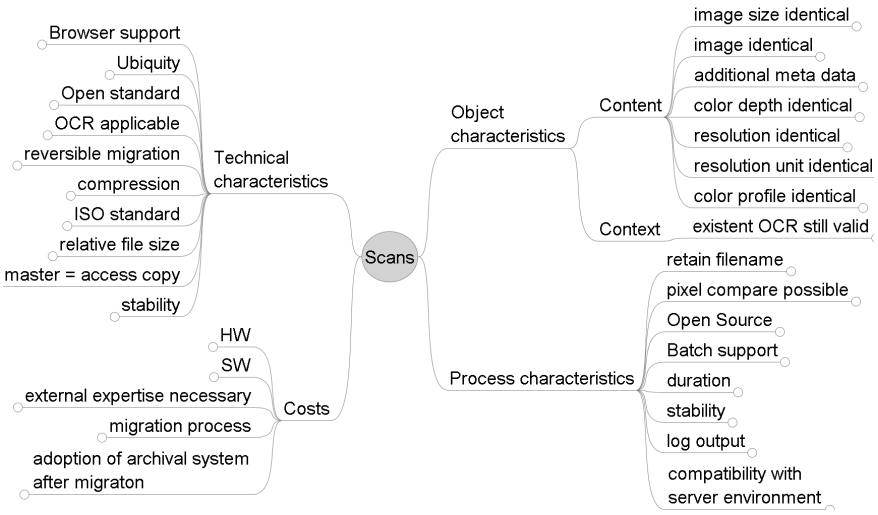


Figure 5.12: Scanned book pages requirements tree

third alternative is rejected due to loss in the image data, which leaves four candidates that are compared using the weighted sum. The sensitivity analysis calculcated automatically in the planning tool reveals in this case that on the level of process criteria, there is a sensitivity to changes in evaluation or weighting. The weighted aggregated utilities of the four alternatives with respect to the requirements branch *Process characteristics* all are between 1.04 and 1.14, and any shifts in the criteria *duration* or *costs* may eventually change the ranking of candidates within the process branch. However, this has no influence on the fact that overall, keeping the status quo is clearly preferred to the other three options; sensitivity analysis shows the robustness of the ranking on the root level. Storage will be monitored and the decision periodically reviewed.

5.4.3 Scanned negatives of aerial photographs

A third evaluation with a very similar scenario was carried out by the Royal Library of Denmark[8], creating a preservation plan for digital safety copies representing original black-and-white cellulose nitrate negatives of aerial photographs stored as TIFF-6 images. Negatives in unstable condition are scanned in a high safety copy quality (1800 ppi, RGB, 16 bit) suitable for eventual replacement of the original material, while negatives in good condition are scanned in standard quality (1800 ppi, Greyscale, 8 bit). Figure 5.13 shows a typical image[9] with about 164 megapixels.

[8] `http://www.kb.dk/en/index.html`

[9] Copyright held by The Royal Library, Denmark.

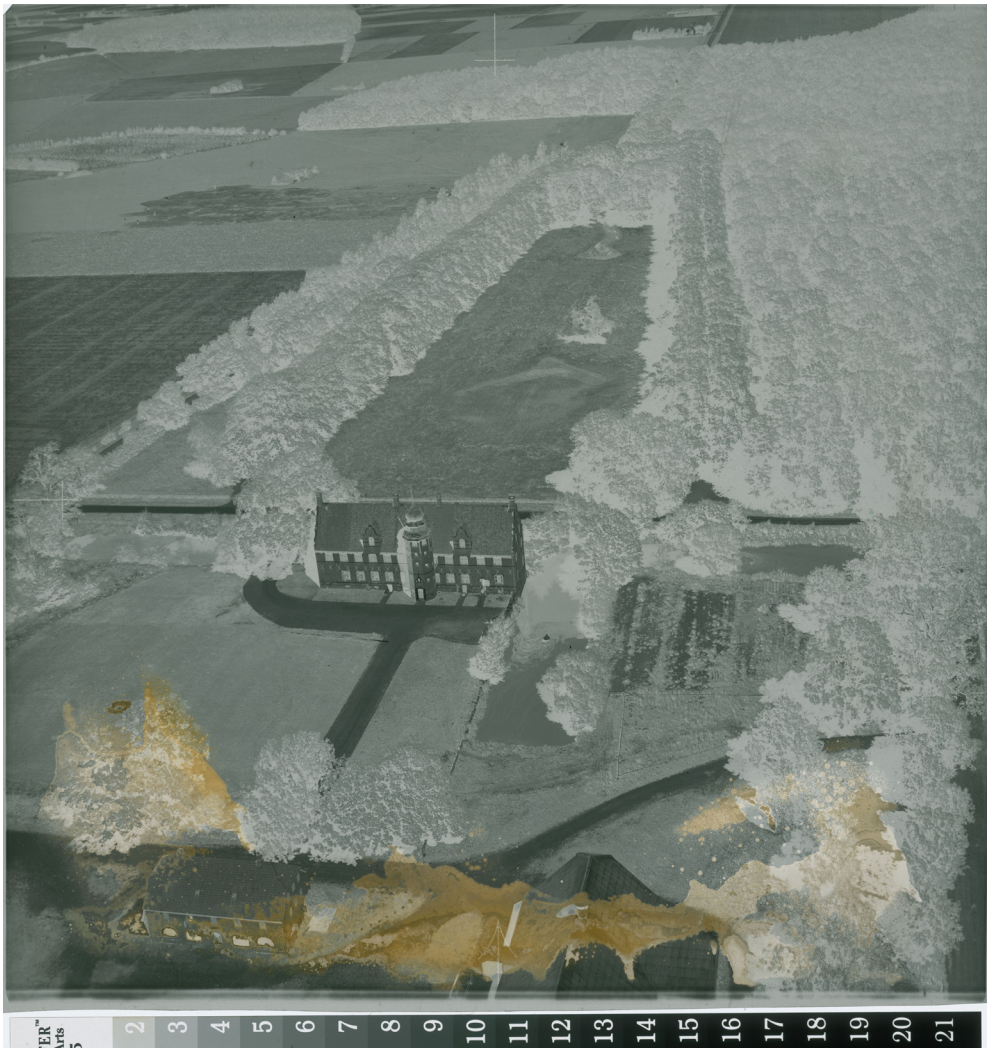
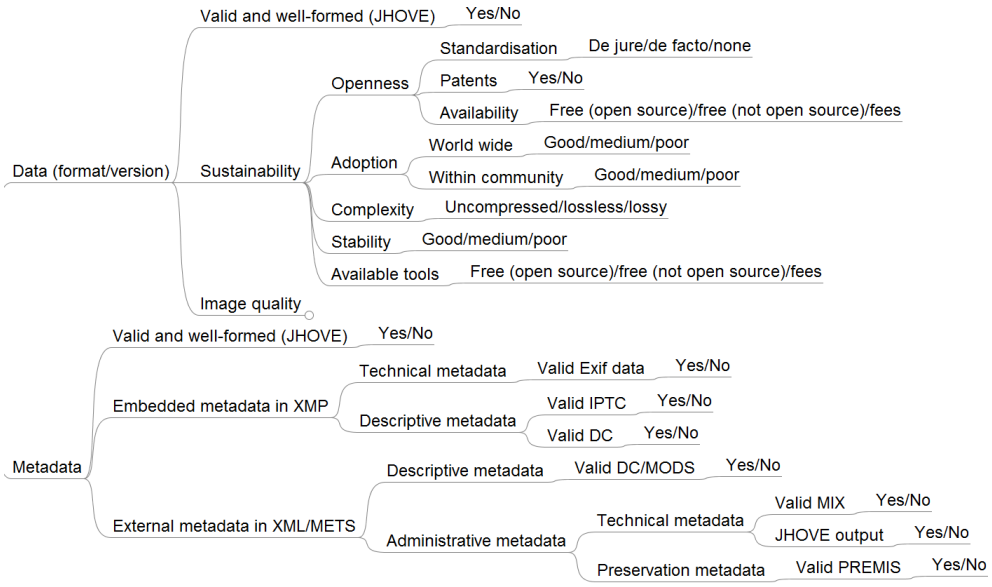
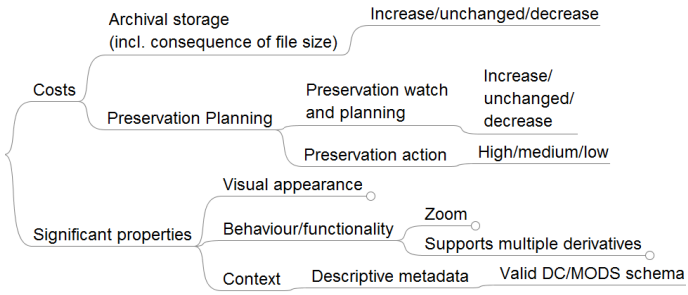


Figure 5.13: Aerial photograph negative scan



(a) Content, Format, Metadata



(b) Costs, Appearance, Behaviour, Context

Figure 5.14: Aerial photographs requirements tree

The rationale for evaluating alternative strategies to storing the large images in TIFF was again motivated by the potential cost savings on archival storage that can be achieved by the smaller file sizes of JP2. The evaluation focused on migration to JP2 and compared ImageMagick as widely available open source tool with the commercial solution LuraWave JP2 CLT (Command Line Tool).

Figure 5.14(a) shows the *object characteristics* branch defined in the study, which is separated into data and metadata. As indicated by the formulation of criteria, the evaluation procedure relied on the output of JHOVE to facilitate semi-automated evaluation of conversion quality. The evaluation values were compared manually and entered into the planning tool, but relied on the properties extracted by JHOVE. The structure of the requirements

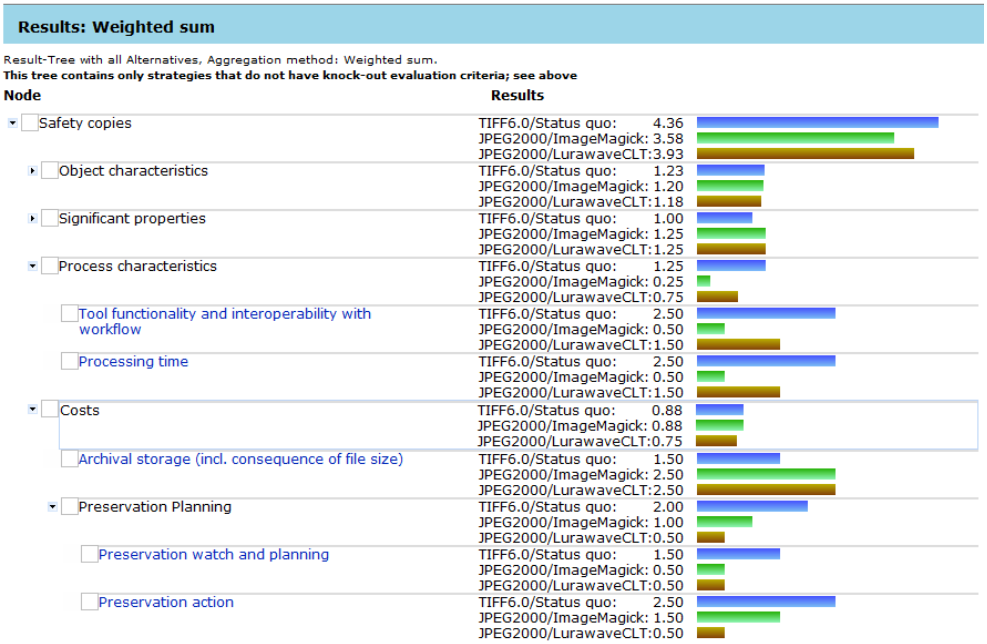


Figure 5.15: Top level results for aerial photographs

Candidate action	Weighted multiplication	Weighted sum
Keep status quo (TIFF-6)	4.14	4.36
Migrate to JP2 (ImageMagick 6.4)	2.89	3.58
Migrate to JP2 (Lurawave JP2 CLT)	3.51	3.93

Table 5.4: Evaluation results for preservation actions on aerial photographs

on object characteristics varies from the often-made distinction between the format and the 'intellectual' properties and instead distinguishes between data (comprising both the format and the content characteristics) and the metadata, while describing the remaining aspects in a separate branch of the tree shown in Figure 5.14(b).

An interesting aspect in the requirements hierarchy is the notion of several aspects of costs that are often neglected. The upper part of Figure 5.14(b) describes expected variations in costs in terms of archival storage (taking into account the file size), but also with respect to expected future efforts for planning and watch. The idea is that certain formats require constant attention and monitoring. While there are no exact estimates of costs – it was deemed infeasible to calculate these costs in exact figures – the directions are seen as useful indications.

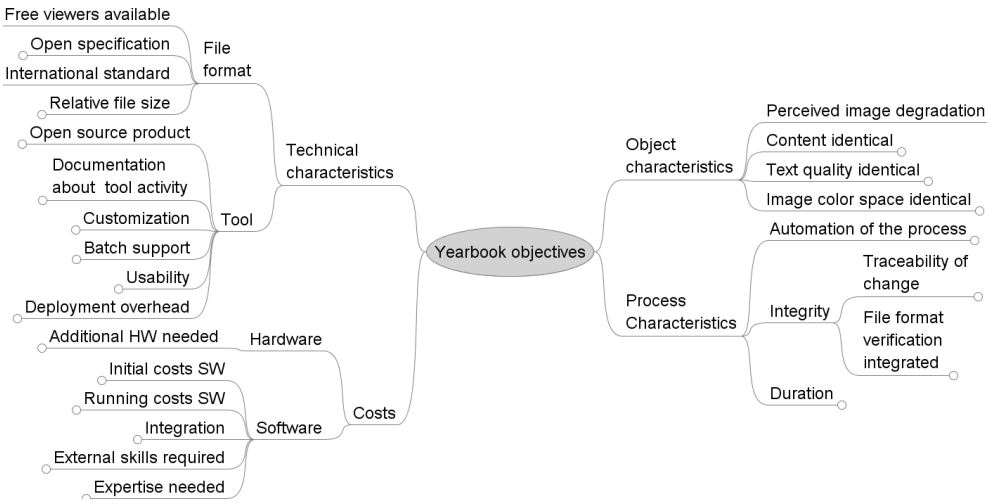


Figure 5.16: Requirements for scanned yearbooks

The final evaluation results are shown in Figure 5.15 and Table 5.4. The analysis reveals a trade-off decision between one-time costs and running costs. TIFF-6 leads to higher storage costs, but wins in terms of watch and planning, and of course leaving the objects unchanged requires very little investment.

5.4.4 Scanned yearbooks

The last case study was conducted with the State and University Library Denmark[10], evaluating the options for preserving a collection of scanned yearbooks published in the years 1965-1989. The images were stored in different versions of the GIF format. The storage costs were not as important, since the data volume was not as high in the previously described studies. The objective tree is shown in Figure 5.16. In contrast to the other three cases, this study was not meant for productive decision making, but purely for evaluation purposes. Analysis and evaluation led to the recommendation to migrate the images to TIFF-6 despite the growth in file size.

5.4.5 Summary

We have discussed a series of case studies that show several important commonalities. They all were analysing preservation actions for scanned images; they all took place in a national library; and they all were evaluating whether a migration to a more suitable format would decrease risks and lower long-

[10] `http://www.statsbiblioteket.dk/english/`

Scenario	Chosen action	Main reasons
80TB scanned newspapers in TIFF5	Migrate to JP2	Storage costs, Standardisation
72TB scanned book pages in TIFF6	Leave unchanged and monitor	Colour profile complications, Lack of JP2 browser support
Aerial photographs in TIFF6	Leave unchanged and monitor	Lack of JP2 browser support, Process costs
Scanned yearbooks in GIF	Migrate to TIFF-6	Format considerations

Table 5.5: Different decisions for preserving scanned images

term costs in return for an acceptable investment, while keeping all significant properties unchanged.

Why did these cases all lead to very distinct conclusions?

Table 5.5 summarises some of the core aspects that differentiate the scenarios. In the first case, storage costs were directly dependent on the file size and substantial; the file format was TIFF version 5, which is not a fully standardised format. Migration to the ISO-standardised lossless JP2 provided the opportunity to lower costs and risks without threatening the content.

In the second case, the cost structures were different, and storage space less of an issue. Moreover, the images were already stored in version 6 of TIFF, which is recognised as an ISO standard. On the other hand, the particularities of the colour profiles embedded in the images made conversion risky and hindered automated quality assurance; thus, a migration would have incurred more costs then it could have saved. In the third case, the images were similarly stored in an ISO-standardised format, and thus leaving the images unchanged was a simple and safe solution. The access costs of creating derivative copies would not have been lowered with the usage of JP2, since current browsers do not natively support JP2, and the costs of migrating to JP2 were thus not considered worth the potential savings. In both cases, a monitoring task has been defined to watch upcoming browser support for JP2, as this may change the preference towards migration to JP2. Finally, in the fourth case, storage requirements were relatively low and the benefit of a standardised format considered enough reason to recommend migration to TIFF-6.

The fact that the analysis of these closely related scenarios led to such different recommendations clearly demonstrates that a preservation action that is optimal in one situation does not necessarily address the problems of another scenario efficiently and effectively. It shows that preservation planning has to take into account the institution-specific preferences and constraints, the peculiarities of the content, and the specific context of each scenario.

It is worth noting that while the decision might be to leave the objects unchanged, this is still a valid and complete preservation plan and vastly different from not defining any action to be taken. On the one hand, a thorough analysis is needed before taking a decision on whether to act or not; on the other hand, the preservation plan contains monitoring conditions that can trigger a re-evaluation due to changed conditions in the future. Trustworthiness as discussed in Section 3.6 requires transparent and well-documented decisions and ongoing management.

5.5 Lessons learned

This section tries to distill the issues discussed in this chapter and draw conclusions from the extensive real-world application experience collected during the past years. We discuss several commonly encountered misperceptions in applying the approach, and thus set the scene for undertaking a critical assessment of the status of the planning approach and tool in the next section.

The effort to conduct an evaluation depends on the types of digital objects, the level of automated measurements that is available, and the experience an organisation has with digital preservation in general and the decision making procedures in particular. Documentation of organisational policies is often scarce; this is beginning to change [SBD+09]. Often, the first time evaluation and selection is carried out, a number of organisational factors have to be established once, which adds to the effort needed for the first decision process. As a general rule, a selection procedure will take at least a few days, involving several stakeholders for the requirements definition and experts for the evaluation.

Based on our experience in the presented and further case studies, and our discussions with users of the planning tool applying the approach to create preservation plans, we observe common misperceptions of a number of concepts and tasks that play a central role in preservation planning.

- **What is a plan: Misdefined scope.** – A common perception is that of a preservation plan defining the steps needed to create and setup a repository, to start running a repository software, or to define very general and high-level rules to follow in operations. While these aspects are certainly relevant and even fundamental to a trustworthy repository, they are considered predecessors. They are necessary precursors of applying the planning approach presented here. Other approaches such as the PLATTER[11] tool for high-level repository planning cover some of these aspects. In contrast, the preservation planning approach

[11] http://www.digitalpreservationeurope.eu/platter/

presented here is focused on defining the actions to be taken to secure access to (a subset of) the objects contained in such a repository.

- **What is a policy: Misdefined assumptions and constraints.** – In relation to the previous issue of scope, we encounter the perception that a plan is a policy or vice versa. For some institutions, a policy might define that all images have to be kept in a certain format. While in some instances this may be feasible, such general assumptions usually cannot be used directly for successful operations. We have seen in the discussions on the image case studies that even in such seemingly simple and obvious cases, there is a substantial variation in each of the scenarios, bringing about subtleties that require specific attention. Given the fact that most scenarios are far more complex than these, it is highly unlikely that general rule-based approaches purely relying on generic policy statements can be sufficient and effective. They do not provide concrete actionable steps such as the selection of a specific component and the definition of parameter settings. Moreover, they do normally not support the provision of traceable evidence demonstrating that the decisions were made based on solid in-depth evaluation and thorough analysis.

  We thus define policies as high-level influence factors, modelling environmental and organisational constraints and explicitly specifying organisational priorities and preferences. Based on these fundamental statements about the decision space, we can then operate productively by creating, monitoring, and updating concrete, actionable plans in the manner described in this thesis.

  Planets has developed a high-level model of institutional policies, which we have started using in the abovementioned case studies. This structured documentation proved to be very valuable in the decision process as it forces stakeholders to explicitly state their preferences and constraints. In fact, the first step of the planning procedure, where a number of basic questions about mandates, policies, and legal obligations are posed, often causes decision makers to stop, take a step back, and properly specify their policies in a structured manner, as they realise that this is a necessary precursor to operational planning.

- **What is a collection: Misdefined scope of a plan.** – The term *collection* is overloaded, especially in the archival and library communities. In many environments, it refers to genres or a subject classification hierarchy. In constrast, in the planning approach, we need to separate objects according to the treatment we can apply to keep them accessible. This leads to common misperceptions. Often, decision makers start the planning procedure by trying to define a plan either for their entire holdings, or for any subset of their holdings that

they refer to as a *collection*, but which contains a variety of objects that require specific, different, treatment.

The question of what to cover within one preservation plan cannot be answered absolutely; it depends on the objects at hand, the usage patterns and access modes, and the actions available for treatment. If an image migration component can be applied to a variety of different formats, it will often be possible to define one plan for a collection of images even if it contains several different image formats. However, sometimes parts of the collection contain specific content that e.g. requires certain access features. For instance, high-resolution aerial photographs may require access modes such as those provided by JP2, where only specific regions of an image are delivered and progressive scanning can work on different dimensions (not just resolution, but also colour depth or regions). In these cases, the requirements that need to be considered for the subset of the collection may imply that a separate plan can deal more efficiently with a particular scenario (such as that formed by the subset of aerial photographs) than one plan covering all image content.

In general, the *collection* should be defined to cover the largest set of objects that presumably can be covered with one preservation action, so that the evaluation can analyse all potential actions and compare them to each other. It may be necessary to return to the point where the collection was specified and split a plan into several parts, each defining the actions to take for a subset of the previuosly defined collection. More sophisticated workflows that are able to characterise objects and apply different actions according to object types can increase the coverage of action components and thus also the efficiency of evaluation.

- **What is a sample object: Wrong selection of samples.** – The definition of the collection immediately leads to the issue of specifying sample content that is representative of this collection. Depending on the complexity of the objects and the variety of technical features within the collection, this stratification is in some cases a complicated question. In-depth collection profiling and analysis is needed to ensure proper stratification of samples. For a collection of electronic documents, for instance, the contained embedded objects will be of interest, as will be the variety of fonts referenced and the question whether some documents contain a change history and whether this history is considered of any relevance.

  Defining representative content has to focus on the technical side of the objects and cover the difference in structural expression of the content, not the variety of the semantic content that the objects represent (such

as different motives shown in digital photographs). However, sometimes the two are hard to separate. In the scanned yearbooks example mentioned above, sample objects of images from different yearbooks were selected, even though they all had the same dimensions and same technical features in terms of the GIF format in which they were stored. The reason was that the original pages were printed on different paper, the scans thus showed variations in sharpness and the histograms, and the library intended to run OCR analysis on the images.

- **What is being evaluated: Misdefined actions.** – Decision makers sometimes focus purely on finding the *best format* for their content. Early plans sometimes compared alternatives such as *Migrate to PDF/A* with *Migrate to TIFF*. However, the target format is just one of the aspects; it cannot be separated from the action path needed to arrive at the target point. Analysis has to include both desirable outcomes of an action, such as requirements on archival formats, and the requirements on the action needed to achieve these outcomes.

  Moreover, different tools will produce outcomes with different characteristics. For example, not all tools migrating to PDF/A will produce standard-compliant output on all input; and some tools will do so more cost-efficiently than others. Migrating to the 'perfect' format is only the optimal solution if there is a tool available that performs well enough. The object of study, i.e. the alternative actions to be evaluated, thus should always consist of an exact definition of the actions under consideration, such as *Migrate all images of the collection to uncompressed JP2 using ImageMagick 6.4*, including specification of the used version, concrete parameter settings and the computing environment it is run in. Similarly, requirements should focus on the actions and the desired outcomes.

- **What is a requirement: Misdefined criteria.** – The requirements definition is the core part of the planning procedure and hence also the most critical, since misdefined requirements may lead to wrong decisions. A very common mistake is the definition of too abstract scales or the inclusion of numerical scales with weakly defined units and measurement procedures. A related issue is the tendency of many stakeholders to think in terms of solutions rather than problems, thus preempting decisions to be made at a later stage. Examples are requirements detailing desired file formats rather than format characteristics when no formal decision has been taken yet, or defining migration requirements when emulation should be considered as well. Yet, requirements must be concerned solely with the *problem space* and not specify solutions. We have discussed these issues at length; however, two specific mistakes are of particular interest.

- **What is being measured: Misdefined scales.** – The specification of significant properties of objects sometimes fail to distinguish between desirable properties of the outcome of applying an action, such as a Boolean criterion *text should be searchable*, and properties that need to be kept unchanged, such as image width. In fact, properties such as image width are often included as a criterion in the tree with a numeric scale, where the measurement unit is set to *pixels*. While this is a correct specification of image width, the *objective* is not image width per se, but the fact that it shall be kept unchanged. The proper specification thus may read *Image width unchanged*, measured on a Boolean scale.
- **What is 'acceptable': Measurements and utility.** – A similar inexactness occurs when a property cannot be measured automatically in sufficient detail (unlike image width). For example, the early case study described in Section 2.4.2 defined criteria for a number of significant properties contained in electronic documents. These criteria described the objective that aspects such as footers, equations, and tables should be kept intact; the scale used was usually *Yes, Acceptable, or No*, stemming from the fact that evaluation had to be done manually due to the lack of automated measurement tools.

  However, the goal underlying our approach is to collect objective measurements on objective scales, and then apply utility functions to model the subjective acceptance thresholds and specifics of the stakeholders. Defining measurement scales that include acceptance mixes the objective and the subjective and makes it almost impossible to reproduce the measurement stage later on. Definition of these scales should instead be explicit about the loss that was encountered and thus strengthen the documentation.

  The question of acceptable loss must not be answered in the measurement phase, but instead in the step of transforming the well-documented measurements into utility values specific to the evaluation scenario. For example, if the integrity of footers is a requirement, the scale that this requirement should be measured on could either consist of a percentage value of footers preserved correctly, an absolute count of footer instances lost per document, or at the very least an ordinal value specification such as *Fully intact, Differences, Severe Losses*. Whether a small loss is considered acceptable or not is a separate decision.

  Looking at the long-term evolvement of plans, the potential effects of ill-specified scales become clear. Consider a case where the policy of an institution changes from accepting the loss of font information, as long as fonts are replaced with similar types, to not

accepting any font replacement. If *fonts* had been evaluated using a scale of *Yes, Acceptable, No*, it would be impossible to change just the utility function, and the complete requirements specification and evaluation procedure would need to be re-run. If the scale instead had at least been *Identical, replacement with font family, Replacement with standard font, Loss of fonts*, it would suffice to change the utility function. The more exact the specification is, the more repeatable become the measurement process and its result.

- **What is important: Overdefined weightings.** – Sometimes decision makers spend a lot of effort on exactly specifying their relative preferences down to the very last hierarchy level of the tree, discussing questions of minute detail. However, it should be noted that the changes in importance factors at low levels of the trees have almost no influence on the final ranking. The key effect that critical low-level criteria have on rejecting alternatives is through the zero utility knockout, which does not depend on the relative weights. Most often, an equal weighting is thus sufficient for the lower levels of the objective tree. The high level priorities, however, should be balanced carefully.

In general it should be noted that there are three steps where an institution influences the evaluation outcome:

1. Requirements definition,
2. Transformation settings, i.e. definition of the utility function, and
3. Importance weighting of requirements.

Requirements definition needs to be complete and along the correct lines of measurement; transformation has to define the acceptable parameter boundaries and establish utility values for each dimension; and the importance factors need to reflect the institutional priorities. At each of these steps, there is a risk of weakly defined and weakly documented assumptions and a corresponding need for thorough analysis, automated quality checks, and tool support.

Summarising these issues, requirements specification, evaluation, and transformation are complex procedures that at first may overwhelm decision makers. The software tool Plato provides considerable support and enables planners to reuse experience of others through a shared knowledge base. Still, the overall complexity of the problem implies that more sophisticated tool support and automation is needed.

5.6 Criticism and gaps

While the case studies discussed here included coaching, there have been several cases of successful planning without coaching. However, the lessons that can be drawn from the extensive real-world experience show the complexities involved in the planning activity and indicate that strong tool support and substantial knowledge is needed to successfully create a preservation plan. This section will discuss the specific issues that we deem essential for improving the applicability of the method and point out potential for improvement of the method and tool.

- There is a lack of structured, informative and reliable information sources.
- Applying the approach of requirements specification and evaluation proved challenging.
- There is no clear way of connecting measurements and requirements and providing ongoing monitoring and re-assessment of quality of service.
- The effort needed to manually create a preservation plan is substantial. This has the effect that for many collections, applying the planning approach is not feasible.

To address these issues, there is a strong need for more sophisticated tool support, automated quality assessment, and proactive recommendation technologies.

5.6.1 Information sources

There is a lack of well-structured information sources that can be queried and integrated automatically. While PRONOM[12] is often cited as a reference source, it does not contain sufficient levels of detail on file formats to truly support automated evaluation and risk assessment. For example, a key factor in evaluating the risk of a format is its viewer support, which can be estimated by counting the number of tools that are able to read the format. While PRONOM in principle provides this information, it contains only a miniscule fraction of the world's file viewers. Other sources such as the Digital Formats Web site[13] represent valuable sources of information, but on a very restricted level of detail. The Global Digital Format Registry[14] and its successor, the Unified Digital Format Registry (UDFR)[15], promise to close

[12] www.nationalarchives.gov.uk/pronom/
[13] http://www.digitalpreservation.gov/formats/index.shtml
[14] http://www.gdfr.info/
[15] http://www.udfr.org/

this gap in the future. However, none of these can be directly used now on a large scale, since the content is fragmented and incomplete.

Not only analysis of file formats, but also discovery of potential preservation actions is a tedious process that is prone to information gaps. Registries holding information about available tools for preserving digital content are being built, but need to be populated and publicly available. Furthermore, significant experience needs to be accumulated and analysed to provide a basis for shortlisting potential alternatives. While Chapter 4 showed that there is considerable progress in this area, the amount of information contained in public registries is often insufficient and still needs to be complemented by manual investigation.

5.6.2 Requirements specification and evaluation

Participants in case studies were all very confident that the requirements in the end captured their real needs. They were very satisfied with the evaluation results and the transparent documentation that results from the planning procedure. However, requirement specification continues to remain the most challenging part of the planning workflow. This is in part due to the fact that for many institutions, this is still a new area, and thus the high-level constraints and influence factors are not yet settled or weakly defined. For example, it is sometimes not entirely clear which standards must be followed and which are just desirable, or how to calculate costs and assess risks.

The definition of significant properties is a technically challenging and complex issue. Considerable progress has been made through early applications of the described approach and in the INSPECT project. A recent discussion summarises the state of research [KP09]. In Plato, a growing knowledge base of significant property trees is being made available, both community-driven and as part of a moderated procedure within the Planets project. Feedback clearly indicates that this greatly supports and eases the planning procedure; however, it incurs the risk that decision makers do not thoroughly analyse their own needs, but instead simply reuse the needs of others.

There is a substantial variation in the definition of significant properties, of performance characteristics, and of measurable properties in general. This also leads to a lack of comparability of results across case studies. The flexibility to express and model specifics of the scenario, which addresses the fundamental need to take these peculiarities into account, carries considerable difficulties. The possibility to model organisational preferences and utilities is essential, but the objective *criteria* shold be standardised, reusable, uniquely identified, and selected from catalogues; and correspondingly, the measurements need to be clearly defined, repeatable, and reproducible.

The evaluation of the criteria is often unclear, and planners report having considerable difficulties to carry out the evaluation procedure. As an

illustration, consider the simple case of evaluating the quality of image migration from TIFF to JP2. Figure 5.17 shows part of the output extracted by JHOVE in a side-by-side comparison as presented by the planning tool during the evaluation stage. While the integration of these extracted properties into the workflow saves considerable time, the complexity of evaluating criteria such as the ones discussed in the case studies by studying the properties of the images and trying to figure out their meaning is overwhelming for many decision makers.

5.6.3 Conceptual links and monitoring support

The conceptual link between influence factors and the impact that changes in these have on decision preferences is a complex and critical problem. Creating and maintaining the conceptual connection between these influence factors and the outcomes of decisions via manual monitoring is a difficult task and a largely unsolved question, and manual evaluation of experiment results can be very time-consuming. The effort needed to analyse objects, requirements and contextual influence factors is in many cases prohibitive. Characterisation tools support the automated comparison of objects; but there is a variety of requirements which cannot be measured automatically yet, and there is no clear way of tracing back measurements to requirements, documenting the measurements in tight integration with the evaluation procedure, and monitoring the continuous operation, the *quality of service*, of deployed preservation plans after the selection and plan creation procedure.

5.6.4 Manual effort

Case studies have shown that the manual effort needed to specify requirements, evaluate alternatives and create a preservation plan is often prohibitive. The case studies described in this chapter each involved several people for about a week, including a planning expert to coach the decision makers. The addressed holdings, however, constitute only a fraction of the institutions' overall content. This has the effect that for many organisations, applying the planning approach to all or even just the most valuable collections is not feasible. It is evident that substantial tool support and automation is needed to decrease the amount of manual involvement and thus make it feasible to create and monitor preservation plans in the large.

5.7 Summary

We have seen in this chapter the practical application of the method and tool presented before, which are increasingly being used for decision making and preservation plan definition by large institutions. We have reported on

Figure 5.17: Plato showing the property trees of one image as extracted by JHOVE from the original TIFF (left) and a derivative JP2 (right)

several case studies creating preservation plans for different types of objects, discussed typical mistakes made, and described lessons learned.

The conclusions drawn show that the concrete requirements evaluation procedure is still weakly defined and particularly effort-intensive. Case studies indicate that manually creating a plan for a significant number of collections is often practically infeasible due to the high costs incurred.

We noted earlier that evidence is an essential precursor to trustworthiness, and that an entity's trustworthiness has to be evaluated in the realistic context of an action. Thorough documentation is needed to ensure reproducibility of evaluation experiments.

The next chapter will show how a large part of the requirements can be automatically measured in controlled experimentation. This not only reduces the effort needed to evaluate components, but also supports trust in the decisions because extensive evidence is produced in a repeatable and reproducible way and documented along with the decision in a standardised and comparable form. It further provides the basis for continuous monitoring of operational preservation plans based on QoS specifications and service level agreements. We claim that these benefits can be achieved for a large fraction of the decision criteria. We will evaluate this claim by quantitatively assessing the coverage of automated measurements with respect to criteria used in real-world decisions.

Chapter 6

Controlled experimentation and automated measurements

This chapter addresses the identified core difficulty of preservation planning. While we have a solid framework for evaluation and decision making, the actual evaluation process is still weakly defined, and it is unclear how measurements can be obtained. Yet to provide a trustworthy, reproducible and repeatable evaluation and selection method and tool that is scalable and supports continuous monitoring, we need substantial and reproducible evidence, which can only be provided by repeatable measurements.

We argue that controlled experimentation and automated measurements can be used to rigorously evaluate preservation components for ranking, selection, and monitoring, since the following two conditions are met:

1. The functionality of components is homogeneous and well-defined.
2. The number of components and instances of the selection problem is sufficiently high to value the additional effort needed to setup a controlled environment and develop the tools and techniques needed for automated measurements.

We will evaluate our claim by analysing real-world case studies and discussing the criteria defined therein for measurability. Hence, our primary evaluation goal compares the number of measurable criteria against those that defy measurement. The main research questions are thus:

1. What categories of criteria have to be evaluated?
2. What entities do we need to measure?
3. How can we obtain the measurements?
4. How many of the criteria can be measured?
5. How large is the effort needed to take measurements?

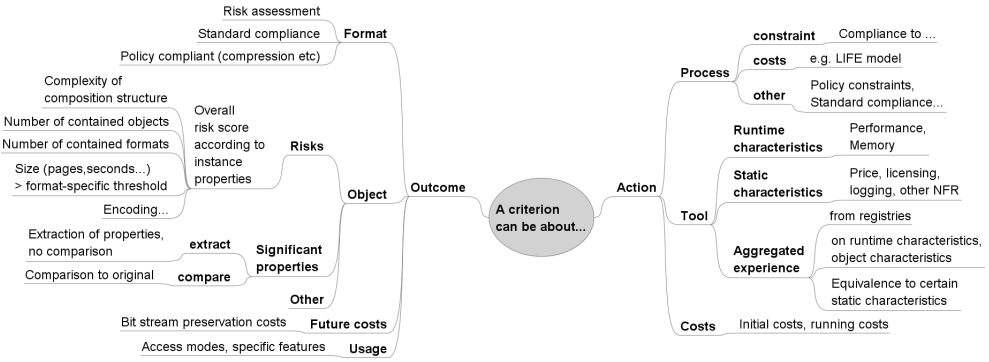


Figure 6.1: Preliminary taxonomy of criteria

6. How can we ensure the measurements are correct?

In Section 6.1, we will first present a taxonomy of criteria that is complete in the sense of covering all influence factors encountered. The subsequent sections discuss the question of measurement for each category of the classification hierarchy and provide examples for automated measurements. Section 6.2 discusses the comparison of object characteristics; Section 6.3 shows extraction of structured data; and Section 6.4 presents a framework for quality-aware migration services, focusing on performance measurement. Section 6.5 shows the automated analysis of information obtained from trusted information sources, and Section 6.6 demonstrates how all the measurement instruments are integrated in the planning tool. Section 6.7 will return to the questions posed above and conduct a quantitative assessment of our claim.

6.1 An evaluation framework

6.1.1 A taxonomy of criteria

To design and evaluate a full-coverage measurement framework for digital preservation, we have created a taxonomy of criteria that differ in the information sources they depend on to obtain measurements. Figure 6.1 shows an early, rather extensive version of the taxonomy that was built after analysing ten of the case studies conducted using the planning approach, containing several hundred criteria. Fundamentally, all criteria requiring measurement refer either to the action, i.e. the component, or the outcome of an action, i.e. a rendering or transformation of a digital object. Risks as a major concern are modelled explicitly, as are significant properties. Tool characteristics are subdivided into runtime, static, and aggregated experience. Format evaluation addresses risk assessment, standards compliance, and policy compliance.

This classification served as an intermediate step, where the goal was to separate categories according to the fundamental source and type of mea-

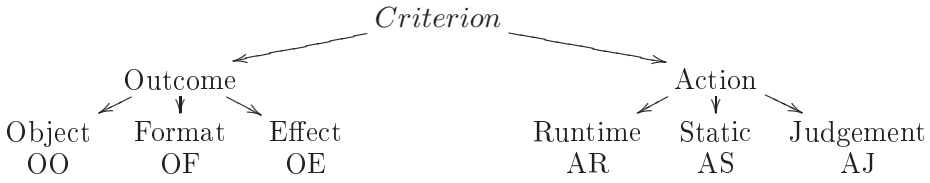


Figure 6.2: Taxonomy of criteria in digital preservation

surement and what entity it needs to be applied on. This led to the final classification depicted in Figure 6.2. The two top level categories *Outcome (O)* and *Action (A)* can be further distinguished to describe general effects of the outcome (OE), such as the expected annual storage costs that result from applying a certain action; criteria describing the format of the objects (OF); and criteria describing the abovementioned significant properties of objects (OO). On the other hand, action components exhibit properties that are static and descriptive in nature (AS), properties that can be measured at runtime (AR), and some properties that depend on judgement (AJ).

The taxonomy is in principle orthogonal to the structure of our hierarchy of goals, i.e. an evaluation goal can be connected to measurable characteristics of different categories. For instance, the general goal of inducing minimal costs may include both the price per object, i.e. per execution, of a component (AS), and runtime characteristics such as resource utilisation (AR) that imply a certain level of hardware expenditures.

We have thus identified the following categories.

1. Properties of the **outcome** of applying a component.

   (a) **Object**. This category entails all desired properties of digital objects. This includes simple properties that are seen as desirable, such as the *searchability* of text documents, and properties that have to be kept unchanged compared to the original object. Properties of the resulting objects, such as the ability to search or edit text documents, need to be measured on the outcome of applying a preservation action. For significant properties that have to be kept intact, the base measures taken on the outcome of the preservation action have to be compared to the base measures obtained from the original object. For example, the criterion *Textual content unchanged* is measured by analysing the original object and the outcome of the preservation action and comparing these for textual equality to get a derived measure on a Boolean scale. We thus obtain this measure by comparing the text content of the original object to the text content of the action result. Sections

6.2 and 6.3 discuss means for measuring and comparing object criteria.

(b) **Format**. This category comprises criteria that specify desirable characteristics of the formats that are used for storing digital content. As a significant portion of the risks to digital content lies in the form of representation and its understandability, this is often a central decision criterion. Typical criteria include standardisation (e.g. *Format is standardised by ISO*), format complexity, or openness of formats. These criteria comprise compliance to institutional policies as well as preferences for low-risk formats; what an institution considers a low risk depends on its risk profile which is modelled in the utility functions. Measurements of these criteria are applied by analysing the format of the outcome and getting additional information on known properties of certain formats from trusted external data sources such as the Pronom Technical Registry[1]. This is described in Section 6.5.

(c) **Effect of outcome**. This refers to any other effects caused by the application of a certain component, such as the storage costs resulting from converting to certain formats with higher compression. Typically, these effects are calculated by organisation-specific models or recognised cost models such as LIFE [ADM+08], based on measures as model inputs. For example, storage costs will depend on organisational cost structures, but strongly correlate with the file size of objects. The file size of the output objects measured in relation to the originals can thus be used as input for a cost model computing the total annual storage costs of a collection.

2. Properties of the components, i.e. the **action** taken.

(a) **Runtime**. This category entails runtime properties of components such as performance and resource utilisation. Measurements need to be taken in a controlled environment. Section 6.4 will present such an environment.

(b) **Static**. Criteria of this category refer to properties of the action components that do not vary per execution run nor show differences when evaluated by different users; i.e., they are not subject to the evaluator's perception and can be determined objectively. These criteria can thus often be obtained from trusted sources. For example, the question whether a component is open source or not should be documented in component registries. Where not found, these criteria need to be evaluated manually with appropriate documentation.

[1] `http://www.nationalarchives.gov.uk/PRONOM/Default.aspx`

(c) **Judgement**. This category is sometimes relevant, but should be kept to a minimum. Usability is a prime example where judgement is necessary. In digital preservation this does not have high influence on the decision, since the components to be evaluated are not to be applied by an end user. In other cases, this has more importance; but in any case, proper documentation of evaluation values is essential.

The obvious question arises if outcomes such as searchability are not simply determined by the format. A simple example reveals that just relying on declared format properties cannot be considered sufficient. Consider the requirement that users want to print out documents, and a collection where no copyright restrictions prohibit this. Migration to PDF/A is a viable option, based on the assumption that PDF formats are well suited for printing. However, certain settings will cause PDF/A documents to restrict printing; and these settings may only be effective when objects with certain properties are used as input to the migration process. To make sure that the requirements are met, we need to verify the possibility of printing on each sample object that we migrate.

The taxonomy is complete in its expressiveness, since it models all relevant entities encountered in the decision process. To validate the expressiveness, the construction of the preliminary taxonomy was followed by a classification of all criteria encountered in all case studies conducted so far. The categories *Other* were only applicable to three criteria (of several hundreds). Close inspection revealed that these criteria were in fact ill-defined and irrelevant to the decision process: They described the legal IPR status of the original digital objects in such a way that it was invariant of the decision process and the actions involved; no potential action could have possibly changed the IPR status of an existing object. (The only way to influence that status would have been to include into the decision process the action of pursuing a legislative act; and in that case, the criteria would have been classified as *output effect (OE)*. We would like to stress that this is slightly beyond the scope of this thesis, and leave the legal analysis of IPR aspects to the lawyers.) We will discuss the distribution of encountered criteria across the final taxonomy in Section 6.7.

6.1.2 Automated measurements

Starting at the classification hierarchy, we analyse how to obtain measurements for each of the identified classes. We develop a family of **Evaluators** that extract and analyse information about objects and components and thus provide an evaluation value for a specific measurable property.

Essentially, evaluators provide characteristics of either objects or actions. A prime example for the first category is risk assessment of objects and ob-

ject formats. Risk assessment for objects has to address two categories of risks: (1) General risks of formats, such as complexity or lack of documentation, and (2) Risks that can apply to objects of a certain kind. For example, Word documents with more than 1000 pages may be much more difficult to preserve than short documents; for plain text files or PDF/A documents, large sizes might be of less concern. Analysing the characteristics of preservation action components, such as measuring the performance of migration tools or services, falls into the second category, actions.

Some of the information that needs to be extracted can be obtained by querying reliable information sources or extracting information from structured data. This mostly applies to documented properties of file formats and actions. Accuracy criteria need to be evaluated by applying measurements on the objects, while runtime properties of the actions have to be measured directly during the experiment. We can thus distinguish several basic types of evaluators, which we will discuss in the next sections.

1. Comparison of digital objects is covered in Section 6.2.
2. Extraction of attributes from structured information sources such as documented metadata schemas will be described in Section 6.3. This includes further characterisation and comparison, since existing characterisation tools such as FITS deliver standardised XML results that need to be analysed.
3. Runtime analysis of action components is described in Section 6.4.
4. Finally, accessing registries that contain trustworthy information is covered in Section 6.5.

6.2 Comparing object characteristics

Validating the content of objects before and after (or during) a preservation action is one of the key questions in digital preservation. Comparators are used for comparing significant properties of objects to validate that the application of a preservation action has not led to a breach of authenticity by destroying or changing a significant characteristic of the original object. To this end, they rely on characterisation tools and services and combine the outputs of these to evaluate changes in the resulting object. In other words, they compute derived comparison measures on base measures using a certain comparison metric.

While a variety of tools are available, this section focuses on the eXtensible Characterisation Languages (XCL) that were described in Chapter 2. XCL is based on canonical descriptions of objects in abstract representations. Consider the migration from PNG to TIFF shown in Figure 6.3. After conversion, the XCDL documents of the original and the transformed

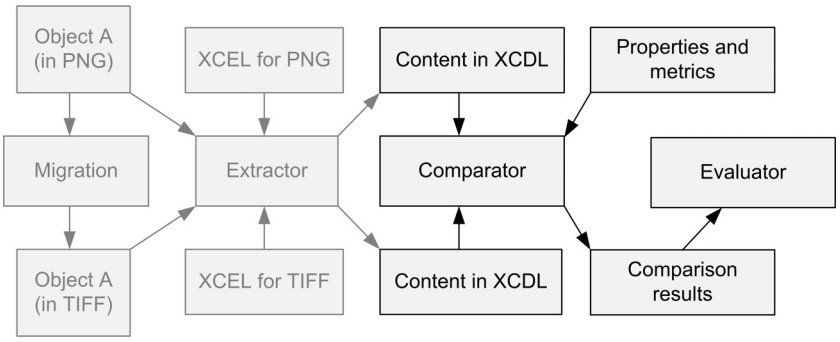


Figure 6.3: Using the comparator on XCDL documents

object can be compared using a comparison component. In its core functionality the comparator loads two XCDL documents, extracts the property sequences and compares them using property-specific definitions of metrics in order to identify degrees of equality between two XCDL documents. The output of the comparison can then be fed into the criteria evaluation. Figure 6.4 shows an exemplary input configuration to the comparison component. It specifies a list of properties to be compared, each with associated metrics that are to be computed. The output of the comparator call is used by the evaluation adaptor in the planning tool.

To allow the usage of this mechanism within the planning procedure, we thus need to connect characteristics and comparison metrics to the requirements and criteria defined in the objective tree. The different layers of this conceptual mapping are outlined in Figure 6.5, which spans the bridge from objects and their characteristics to overall goals and how they can be broken down to more precise requirements and measurable criteria. The two trees need to be modelled in such a way that they can be connected; furthermore, comparison metrics and mapping structures are necessary to support the quantified and automated evaluation of criteria.

While XCL strives to create a canonical representation of objects by defining a direct mapping between formats and abstract representations in the extraction languages, an alternative strategy is to directly look at interpretations of the objects as produced by tools that are assumed to be reliable. We have integrated commonly used standard tools such as ImageMagick *compare*[2].

Table 6.1 lists the available distance metrics and their meanings. This light-weight strategy has the advantage of being very flexible and extensible, but has to be applied carefully: When migrating with ImageMagick, for instance, it would be naive to assume that ImageMagick's own compare tool would recognise errors introduced by the conversion, since both operations

[2] `http://www.imagemagick.org/script/compare.php`

```
<coco ...>
    <compSet>
    <property id="20" name="colourSpaceName">
            <metric id="1" name="equal"/>
    </property>
    <property id="41" name="gammaValueRGB">
            <metric id="1" name="equal"/>
    </property>
        <property id="7" name="normData">
           <metric id="11" name="hammingDistance"/>
           <metric id="20" name="RMSE"/>
        </property>
        <property id="2" name="imageHeight">
            <metric id="1" name="equal"/>
        </property>
        <property id="151" name="bitsPerSample">
            <metric id="1" name="equal"/>
            <metric id="2" name="intDiff"/>
        </property>
        <property id="10" name="backgroundColour">
            <metric id="1" name="equal"/>
        </property>
        <property id="18" name="compression">
            <metric id="1" name="equal"/>
        </property>
        <property id="8" name="gamma">
            <metric id="1" name="equal"/>
            <metric id="2" name="intDiff"/>
            <metric id="10" name="percDeviation"/>
        </property>
            <property id="9" name="orientation">
            <metric id="1" name="equal"/>
        </property>
        <property id="22" name="resolutionUnit">
            <metric id="1" name="equal"/>
        </property>
        <property id="23" name="resolutionX">
            <metric id="1" name="equal"/>
            <metric id="3" name="ratDiff"/>
            <metric id="10" name="percDeviation"/>
        </property>
       ...
    </compSet>
</coco>
```

Figure 6.4: Comparator configuration example

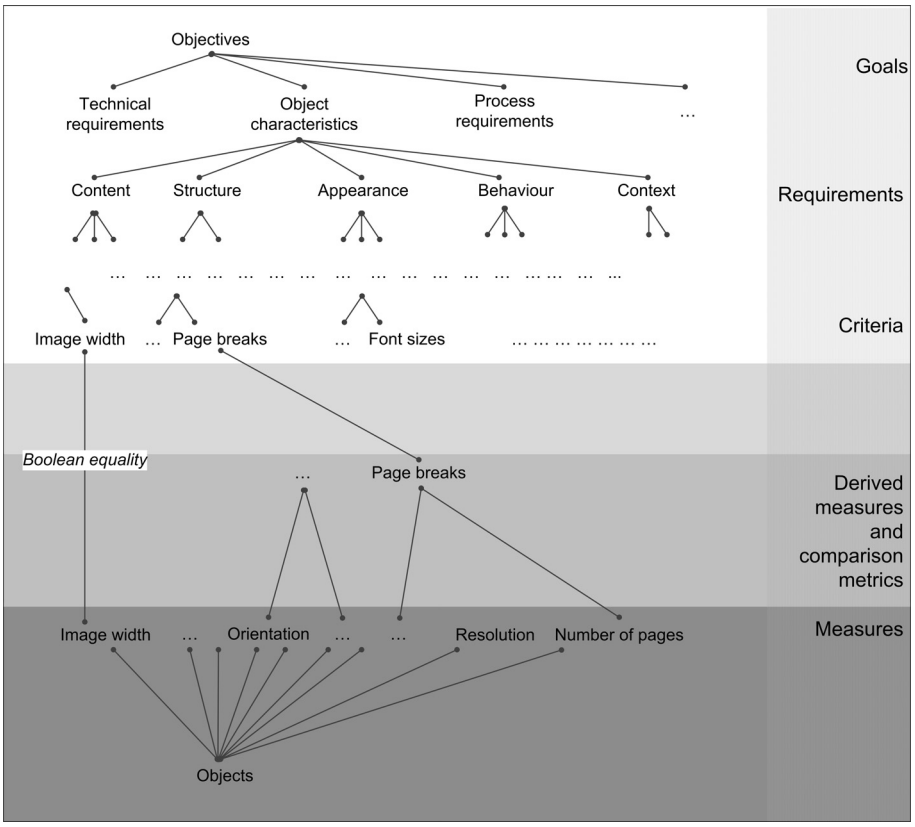


Figure 6.5: Connecting object properties to objectives and criteria

are based on the same set of format interpreters.

It can be seen that the currently deployed object evaluators mostly focus on the rather simple case of images. More sophisticated comparison tools, however, can be integrated easily into the framework, as described in Section 6.6.

6.3 Extracting structured data

Structured data are a common source for information in preservation planning and range from descriptive documentation of action components to dynamically extracted metadata descriptors and experiment documentation. Characterisation of objects also partly belongs to this category, since many characterisation tools produce XML metadata.

While some of the descriptors are format-independent, others differ with the type of objects and even the format involved. Furthermore, some of the properties need only to be extracted, such as wellformedness of a format; with

Abbr.	Metric	Description
AE	Absolute Error	The number of different pixels (0 means identical images). This value can be thresholded to only count pixels that have a difference larger then a specified threshold.
PAE	Peak Absolute Error	The highest difference of any single pixel.
PSNR	Peak Signal to Noise Ratio	The ratio of mean square difference to the maximum mean square that can exist between any two images, expressed as a decibel value. The higher the PSNR, the closer the images are, with a maximum difference occurring at 1.
MAE	Mean Absolute Error	Average over all pixels
MSE	Mean Squared Error	Averaged squared error distance
RMSE	Root mean squared error	Identical to *sqrt(MSE)*.

Table 6.1: Distance metrics computed by ImageMagick *compare*

others, we will need to compare a transformed object against the original. This can prove challenging when the function transforming the model from input format to output format does not provide a homomorphic mapping (which is often the case).

On a metadata level, considerable standardisation has been achieved in some domains by initiatives such as NISO [ANS06]. The characterisation tool FITS[3] (File Information ToolSet) consolidates the output of several other tools such as DROID[4], JHOVE[5] and the ExifTool[6], and provides a normalised XML output. It is furthermore being extended continuously and thus serves well as an integration tool and source of information.

Figures 6.6 and 6.7 show excerpts of two typical XML documents produced by FITS. As shown in the upper part of Figure 6.6, the extraction results of several tools are included. The lower part of Figure 6.6 lists some technical descriptors of the image in question as extracted by the ExifTool: The compression scheme is *deflate/inflate*, image width is 919 pixels, and image height is 615 pixels. Every value is annotated with a provenance information stating which tool extracted the value. The status *SINGLE_RESULT* indicates that were were no conflicts detected between the results of different extraction tools used during the characterisation process.

Figure 6.7 lists only the metadata fragment of the FITS description of a TIFF image. All values were obtained by JHOVE, which shows gaps in terms of PNG support, but has a powerful TIFF extraction module. Apart from image dimensions and compression, we obtain values on properties such as sampling frequencies or colour profiles.

[3] `http://code.google.com/p/fits/`
[4] `http://sourceforge.net/projects/droid/`
[5] `http://hul.harvard.edu/jhove/`
[6] `http://www.sno.phy.queensu.ca/~phil/exiftool/`

```
<fits ...>
  <identification status="SINGLE_RESULT">
    <identity format="Portable Network Graphics" mimetype="image/png">
      <tool toolname="file utility" toolversion="4.26" />
      <tool toolname="Exiftool" toolversion="7.74" />
      <tool toolname="Droid" toolversion="3.0" />
      <tool toolname="ffident" toolversion="0.2" />
      <version toolname="Droid" toolversion="3.0">1.0</version>
      <externalIdentifier toolname="Droid" toolversion="3.0"
       type="puid">fmt/11</externalIdentifier>
    </identity>
  </identification>
  <fileinfo>...</fileinfo>
  <filestatus />
  <metadata>
    <image>
      <compressionScheme toolname="Exiftool" toolversion="7.74"
       status="SINGLE_RESULT">Deflate/Inflate</compressionScheme>
      <imageWidth toolname="Exiftool" toolversion="7.74"
       status="SINGLE_RESULT">919</imageWidth>
      <imageHeight toolname="Exiftool" toolversion="7.74"
        status="SINGLE_RESULT">615</imageHeight>
    </image>
  </metadata>
</fits>
```

Figure 6.6: FITS description of a PNG image

```
  <metadata>
    <image>
      <byteOrder toolname="Jhove" toolversion="1.3"
        status="SINGLE_RESULT">little-endian</byteOrder>
      <compressionScheme toolname="Jhove" toolversion="1.3">
       Uncompressed</compressionScheme>
      <imageWidth toolname="Jhove" toolversion="1.3">919</imageWidth>
      <imageHeight toolname="Jhove" toolversion="1.3">615</imageHeight>
      <colorSpace toolname="Jhove" toolversion="1.3">RGB</colorSpace>
      <referenceBlackWhite toolname="Jhove" toolversion="1.3"
       status="SINGLE_RESULT">0.0 255.0 0.0 255.0 0.0 255.0</referenceBlackWhite>
      <orientation toolname="Jhove" toolversion="1.3">normal*</orientation>
      <samplingFrequencyUnit toolname="Jhove"
        toolversion="1.3">inches</samplingFrequencyUnit>
      <xSamplingFrequency toolname="Jhove" toolversion="1.3">0</xSamplingFrequency>
      <ySamplingFrequency toolname="Jhove" toolversion="1.3">0</ySamplingFrequency>
      <bitsPerSample toolname="Jhove" toolversion="1.3">8 8 8</bitsPerSample>
      <samplesPerPixel toolname="Jhove" toolversion="1.3">3</samplesPerPixel>
      <scanningSoftwareName toolname="Jhove"
       toolversion="1.3">IrfanView</scanningSoftwareName>
    </image>
  </metadata>
```

Figure 6.7: FITS/JHOVE metadata fragment of a TIFF image

Property	Scale	XPath expression
Format valid	Boolean	`/fits:valid[@status='SINGLE_RESULT']/text()`
Format well-formed	Boolean	`/fits:well-formed[@status='SINGLE_RESULT']/text()`
Compression scheme	Nominal	`//fits:compressionScheme/text()`
Image width	Integer (pixel)	`//fits:imageWidth/text()`
Image height	Integer (pixel)	`//fits:imageHeight/text()`
Colour space	Nominal	`//fits:colorSpace/text()`
Bits per sample	Integer	`//fits:bitsPerSample/text()`
Samples per pixel	Integer	`//fits:samplesPerPixel/text()`

Table 6.2: Example properties extracted by FITS

Table 6.2 shows some examples of properties and their extraction paths. The aim of the FITS project is to homogenise as much of the output as possible; the user can further influence the normalisation procedure by defining preferences and rules. For example, it is possible to define prioritisation sequences where it is known that certain tools are more reliable on specific formats than others.

A different example of evaluators are *compliance checks*, where a comparison is made between a declared desirable outcome and the actual outcome. For example, a migration tool may declare its output as being TIFF-6; but this claim has to be validated in the experiments, where a compliance check makes sure that the produced output actually conforms to this declaration by identifying the target format and comparing it against the claimed output format. Figure 6.8 shows a code fragment of an evaluator function deployed in the planning tool that checks format conformance by comparing the declared format identifier to the actual format identifier extracted from the target object.

```
FormatInfo info = alternative.getAction().getTargetFormatInfo();
String puid = info.getPuid(); // Pronom Unique Identifier
String fitsText = extractor.extractText(fitsDocResult,
                "//fits:externalIdentifier[@type='puid']/text()");
return identicalValues(puid, fitsText, criterion.getScale());
```

Figure 6.8: Validating file format conformance

6.4 Quality-aware migration services

While the last sections have discussed measurements for static criteria and objects, this section presents an approach to measuring dynamic runtime

properties of preservation actions. The minimal Migration and Emulation Engine (MiniMEE), described in [BKK+09a] and [BKK+09b], monitors migration components in a controlled environment and thus provides quality-aware component execution.

In Chapter 4, we have presented an overview of a service oriented architecture and a framework for the integration of migration and emulation with selection and planning. We demonstrated that the evaluation procedure greatly benefits from the flexibility and agility provided by service orientation. The integration of heteregoneous systems across platforms through interoperable standards and thus the quick access to functions provided by remote services ease the evaluation and remove the burden of laboursome installation procedures from the planning workflow.

However, these architectures do not solve the issue of measuring the quality of the actions under consideration. Measuring dynamic quality attributes of web services is inherently difficult due to the very virtues of service-oriented architectures: The late binding and flexible integration ideals ask for very loose coupling, which often implies that little is known about the actual quality of services (QoS) and even less about the confidence that can be put into published service metadata, particularly QoS information.

Different aspects of performance measurement and benchmarking of web services have been analysed in literature. As described in [PRD07], there are four principal methods of QoS measurement from the technical perspective.

- *Provider-side instrumentation* has the advantage of access to a known implementation. Dynamic attributes can be computed invasively within the code or non-invasively by a monitoring device.
- *SOAP Intermediaries* are intermediate parties through which the traffic is routed so that they can collect QoS-related criteria.
- *Probing* is a related technique where a service is invoked regularly by an independent party which computes QoS attributes.
- *Sniffing* monitors the traffic on the client side and thus produces consumer-specific data.

The total, round-trip-time performance of a web service is composed by a number of factors such as network latency and web service protocol layers. Measuring only the round-trip performance gives rather coarse-grained measurements. On the other hand, network latencies are hard to quantify, and the run-time execution characteristics of the software that is exposed as a service are an important component of the overall performance. Different levels of granularity can be defined for performance-related QoS; some authors distinguish up to 15 components [WW05]. For the scope of our work, we refer to the 8 components defined in [PRD07]:

1. *Processing time* on the server;
2. *Wrapping time*, needed to marshal and unmarshal of XML structures;
3. *Execution time*, which comprises wrapping and processing;
4. *Latency*, i.e. the time needed for a message to travel the network;
5. *Response time*, i.e. the elapsed time until a response is received for a given request;
6. *Round-trip time*, i.e. the total time needed to complete a service call on the client;
7. *Throughput*, i.e. the number of requests that can be completed by a service in a given amount of time; and
8. *Scalability* as the ability for parallel request processing.

As we discussed in Section 2.5, our evaluation and selection scenario shows some similarities to the general web service selection problem. However, the service instances that are measured are used mainly for experimentation; once a decision is taken to use a specific tool, based on the experimental evaluation through the web service, it might be even possible to transfer either the data to the code or vice versa, to achieve optimum performance for truly large-scale operations on millions of objects.

The implications are that

1. Monitoring the round-trip time of service consumption at the client does not yield sufficient details of the runtime characteristics;
2. Client-side measurement is not very valuable, also because some of the main parameters determining it, such as the network connection to the service, are negotiable and up to configuration and production deployment;
3. Provider-side runtime characteristics such as the memory load produced by executing a specific function on the server are of high interest.

We hence focus on measuring the processing time and memory load of the actual service execution on the provider-side. Other aspects such as wrapping time, latency, throughput and scalability will depend mainly on the target architecture where the selected component shall be deployed in the integration phase, and thus are not contributing to our evaluation needs.

Memory and CPU load of processes are often measured by invasive binary code instrumentation [NS07a] as supported by frameworks such as Valgrind[7],

[7] `http://valgrind.org/`

or by non-invasive monitoring, where the code of the application to be measured is not changed. The latter is much more appealing in this context, since it does not require access to the original code and allows a generic, scalable measurement architecture.

In this section, we present a generic and extensible architecture and framework for non-invasive provider-side service instrumentation. Our framework enables the automated monitoring of different categories of components and provides integrated QoS information. The invoked components are transparently wrapped by a flexible combination of dynamically configured monitoring engines that are each able of measuring specific properties of the monitored piece of software. We present a reference implementation that measures the performance of migration tools and instruments the corresponding services on the provider side. We further demonstrate the performance monitoring of a variety of components ranging from native C++ applications and Linux-based tools to Java applications, and discuss the results of our experiments.

6.4.1 Monitoring framework

Figure 6.9 shows a simplified abstraction of the core elements of the monitoring design and their relations. The key elements are `Services` and `Engines` which are contained in a `Registry`. Each `Engine` specifies which aspects of a service it is able to measure in its `MeasurableProperties`. The property definition includes the scale and applicable metrics for a property, which are used for creating the corresponding `Measurements`.

Each `Engine` is deployed on a specific hardware `Environment` that shows a certain performance. This performance is captured by the score of a `Benchmark` which is a specific configuration of services and `Benchmark Data`, aggregating measurements over these data to produce a representative *score* for an environment. The benchmark scores of the engines' environments are provided to the clients as part of the service execution metadata and can be used to normalise performance data of migration services running on different hardware platforms.

The `Services` contained in a registry are not invoked directly, but run inside a monitoring engine to enable performance measurements. This monitoring accumulates `Experience` for each service, which is collected in each successive call to a service and used to aggregate information over time. It thus enables continuous monitoring of performance and migration quality.

`CompositeEngines` are a flexible form of aggregating measurements obtained in different monitoring environments. This type of engine dispatches the service execution dynamically to several engines to collect information. This is especially useful in cases where measuring code in real-time actually changes the behaviour of that code. For example, measuring the memory load of Java code in a profiler usually results in a much slower performance,

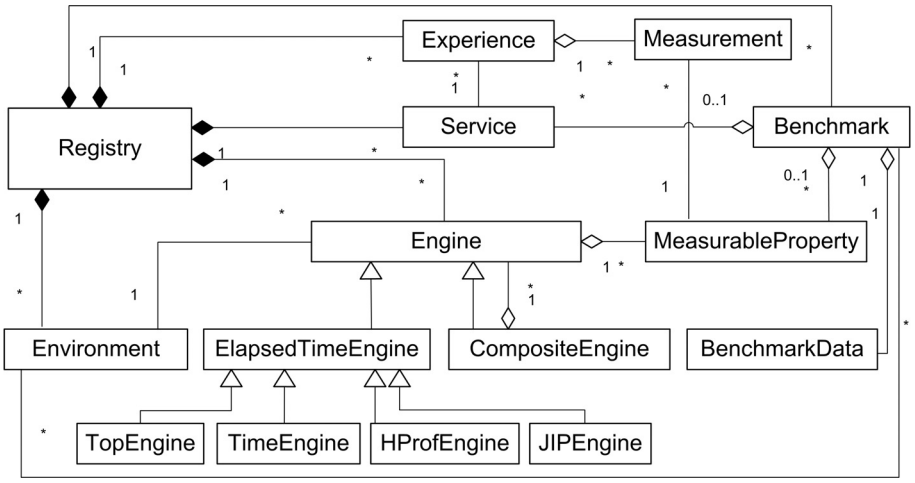


Figure 6.9: Core elements of the monitoring framework

so that simultaneous measurements of memory load *and* execution speed lead to skewed results. Fortunately, in this case there is a way around this uncertainty relation – separating the measurements into different, independent calls leads to correct results.

The bottom of the diagram illustrates some of the currently deployed performance monitoring engines.

1. The `ElapsedTimeEngine` is a simple default implementation measuring elapsed (wall-clock) time.

2. The `TopEngine` is based on the Unix tool *top*[8] and used for measuring the memory load of wrapped applications installed on the server.

3. The `TimeEngine` uses the Unix call *time*[9] to measure the CPU time used by a process.

4. Monitoring the performance of Java tools is accomplished by a combination of the `HProfEngine` and `JIPEngine`, which use the *HPROF*[10] and *JIP*[11] profiling libraries, for measuring memory usage and timing characteristics, respectively.

Additional engine configurations can be added dynamically at runtime. Notice that while the employed engines focus on performance measurement, in principle any category of dynamic QoS criteria can be monitored and benchmarked.

[8] `http://unixhelp.ed.ac.uk/CGI/man-cgi?top`
[9] `http://unixhelp.ed.ac.uk/CGI/man-cgi?time`
[10] `http://java.sun.com/developer/technicalArticles/Programming/HPROF.html`
[11] `http://jiprof.sourceforge.net/`

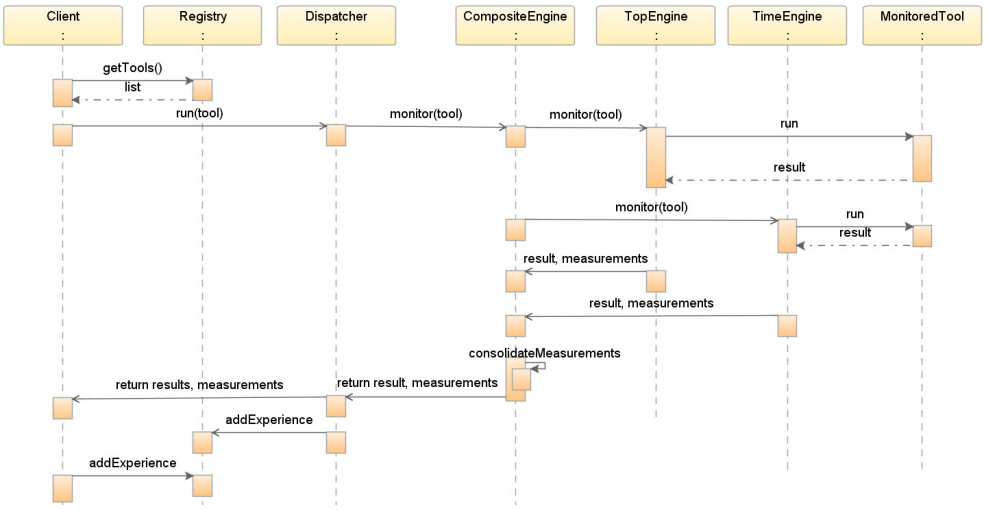


Figure 6.10: Exemplary interaction between the core monitoring components

Figure 6.10 illustrates an exemplary simplified flow of interactions between service requesters, the registry, the engines, and the monitored components, in the case of a composite engine measuring the execution of a component through the Unix tools *time* and *top*. The composite engine collects and consolidates the data; both the engine and the client can contribute to the accumulated experience of the registry. This allows the client to add round-trip information, which can be used to deduct network latencies, or quality measurements computed on the result of the consumed service.

6.4.2 Performance measurement

Measuring run-time characteristics of components on different platforms has always been difficult due to the many peculiarities presented by each tool and environment. The most effective way of obtaining exact data on the behaviour of code is instrumenting it before [NS07a] or after compilation [LB94]. However, efficiency, flexibility and non-intrusiveness are essential requirements in our application context, and access to the source code itself is often not even possible. Hence we use non-invasive monitoring by standard tools for a range of platforms. This provides reliable and repeatable measurements that are exact enough for our purposes, while not necessitating access to the code itself. In particular, we currently use a combination of the following tools for performance monitoring.

- *Time.* – The unix tool `time` is the most commonly used tool for measuring actual processing time of applications, i.e. CPU time consumed by a process and its system calls. However, while the timing is very

precise, the major drawback is that memory information is not always available on all platforms. Depending on the implementation of the `wait3()` command, memory information is reported zero on many environments[12].

- *Top.* – This standard Unix program is primarily aimed at continuos monitoring of system resources. While the timing information obtained is not as exact as the `time` command, `top` measures both CPU and memory usage of processes. We gather detailed information on a particular process by starting `top` in batch mode and continually logging process information of all running processes to a file. After the process to be monitored has finished asynchronously (or timed out), we parse the output for performance information of the monitored process.

  In principle, the following process information provided by top can be useful in this context.

  - Maximum and average *virtual memory* used by a process;
  - Maximum and average *resident memory* used;
  - The *percentage of available physical memory* used; and
  - The *cumulative CPU time* the process and its dead children have used.

  Furthermore, the *overall CPU state* of the system, i.e. the accumulated processing load of the machine, can be useful for detailed performance analysis and outlier detection.

As many processes actually start child processes, these have to be monitored as well to obtain correct and relevant information. For example, when using ImageMagick *convert*, the costly work is in some cases not directly performed by the convert process but by one of its child processes, such as GhostScript. Therefore we gather all process information and aggregate it.

A large number of tools and libraries are available for profiling Java code.[13] The following two open-source profilers are currently deployed in our system.

- The *HProf* profiler is the standard Java heap and CPU profiling library. While it is able to obtain almost any level of detailed information wanted, its usage often incurs a heavy performance overhead. This overhead implies that measuring both memory usage and CPU information in one run can produce very misleading timing information.

[12] `http://unixhelp.ed.ac.uk/CGI/man-cgi?time`
[13] `http://java-source.net/open-source/profilers`

Exp.	Files	File sizes	Total input volume	Tool	Engines
1	110 JPEG images	Mean: 5,10 MB Median: 5,12 MB Std dev: 2,2 MB Min: 0,28 MB Max: 10,07MB	534 MB	ImageMagick conversion to PNG	Top, Time
2	110 JPEG images	Mean: 5,10 MB Median: 5,12 MB Std dev: 2,2 MB Min: 0,28 MB Max: 10,07MB	534 MB	Java ImageIO conversion to PNG	HProf, JIP
3	110 JPEG images	Mean: 5,10 MB Median: 5,12 MB Std dev: 2,2 MB Min: 0,28 MB Max: 10,07MB	534 MB	Java ImageIO conversion to PNG	Top, Time
4	312 JPEG images	Mean: 1,19 MB Median: 1,08 MB Std dev: 0,68 MB Min: 0,18 MB Max: 4,32MB	365MB	GIMP conversion to PNG	Top, Time
5	312 JPEG images	Mean: 1,19 MB Median: 1,08 MB Std dev: 0,68 MB Min: 0,18 MB Max: 4,32MB	365MB	Java ImageIO conversion to PNG	HProf, JIP
6	56 WAV files	Mean: 49,6 MB Median: 51,4 MB Std dev: 12,4 MB Min: 30,8 MB Max: 79,8 MB	2747MB	FLAC unverified conversion to FLAC, 9 different quality/speed settings	Top, time
7	56 WAV files	Mean: 49,6 MB Median: 51,4 MB Std dev: 12,4 MB Min: 30,8 MB Max: 79,8 MB	2747MB	FLAC verified conversion to FLAC, 9 different quality/speed settings	Top, time

Table 6.3: Experiments

- In contrast to HProf, the *Java Interactive Profiler (JIP)* incurs a low overhead and is thus used for measuring the timing of Java tools.

Depending on the platform of each tool, different measures need to be used; the monitoring framework allows for a flexible and adaptive configuration to accomodate these dynamic factors. Section 6.4.3 discusses the relation between the monitoring tools and which aspects of performance information we generally use from each of them. Where more than one technique needs to be used for obtaining all of the desired measurements, the composite engine described above transparently separates the actual execution of the component to be monitored into distinct calls and aggregates the performance measurements.

6.4.3 Experiments

We run a series of experiments comparing a number of migration components for different types of content on benchmark data. The experiments' purpose is to evaluate different aspects of both the components and the engines themselves:

1. *Comparing performance measurement techniques.* To analyse the unavoidable variations in the measurements obtained with different mon-

itoring tools, and to validate the consistency of measurements, we compare the results that different monitoring engines yield when applied to the same components and data.

2. *Image conversion tools.* The purpose of the system in our application context is the comparative evaluation of candidate components. Thus we compare the performance of image migration tools on benchmark content.

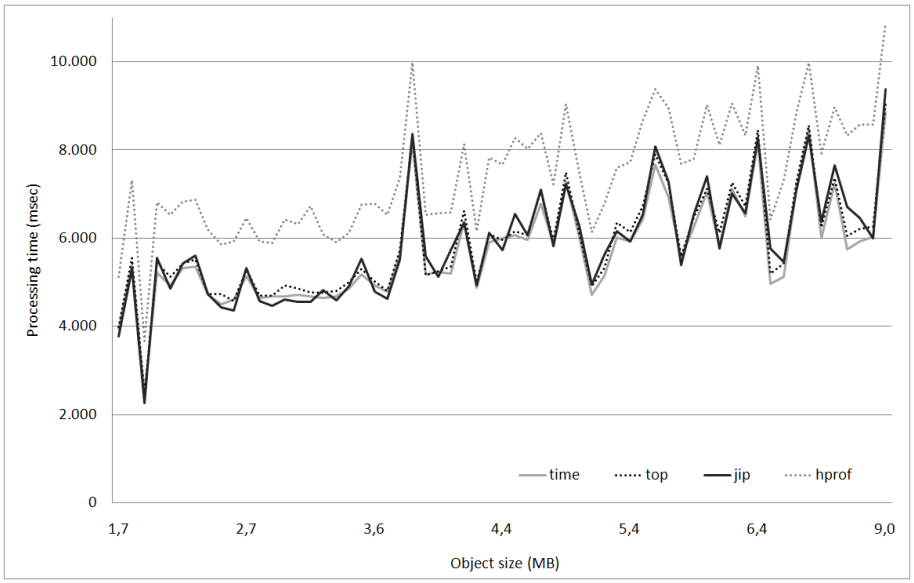
3. *Accumulating average experience on component behaviour.* To evaluate the accumulation of QoS data about each service, we analyse average throughput and memory usage of different tools and how the accumulated averages converge to a stable value.

4. *Tradeoffs between different quality criteria.* In certain scenarios, a trade-off decision has to be made between different quality criteria, such as compression speed versus compression rate. We run a series of tests with continually varying settings on a sound migration component and describe the resulting trade-off curves.

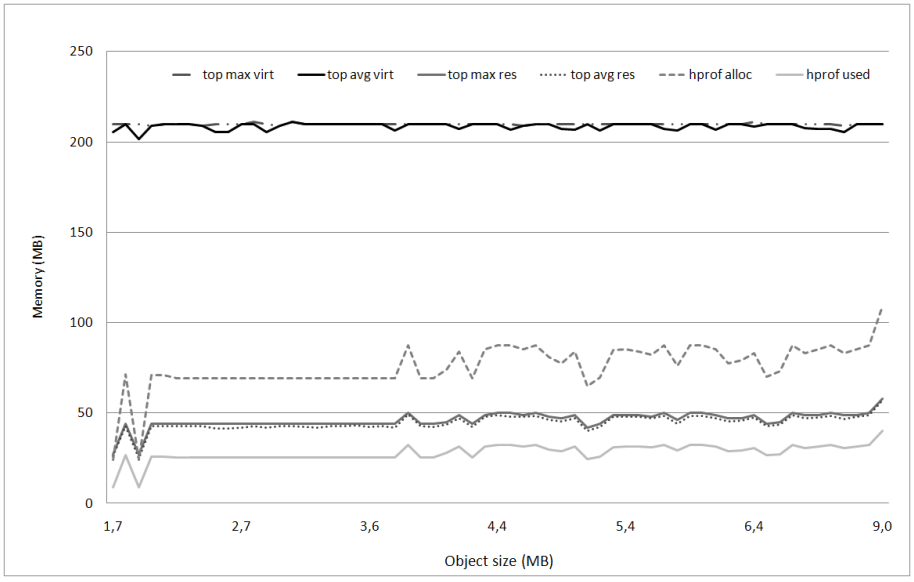
Table 6.3 shows the experiment setups and their input file size distribution. Experiment results in this section are given for a Linux machine running Ubuntu Linux 8.04.2 on a 3 GHz Intel Core 2 Duo processor with 3GB memory. Each experiment was repeated on at least one other applicable machine to verify the consistency of the results obtained.

Measurement techniques

The first set of experiments compares the exactness and appropriateness of measurements obtained using different techniques and compares these values to verify the consistency of measurements. We monitor a Java migration component using all available engines on a Linux machine. Figure 6.11 shows measured values for a random subset of the total files, sorted by size, to visually illustrate the variations between the engines. In Figure 6.11(a), the processing time measured by top, time, and the JIP profiler are generally very consistent across different runs, with an empirical correlation coefficient of 0.997 and 0.979, respectively. Running HProf on the same files consistently produces much longer execution times due to the processing overhead incurred by profiling the memory usage. Figure 6.11(b) depicts memory measurements for the same experiment. The virtual memory assigned to a Java component depends largely on the settings used to execute the JVM and thus is not very meaningful. While the resident memory measured by Top includes the JVM and denotes the amount of physical memory actually used during execution, HProf provides figures for the memory used and allocated within the JVM. Which of these measurements are of interest in a specific component selection scenario depends on the integration pattern.



(a) Monitoring time



(b) Monitoring memory

Figure 6.11: Comparison of the measurements obtained by different techniques.

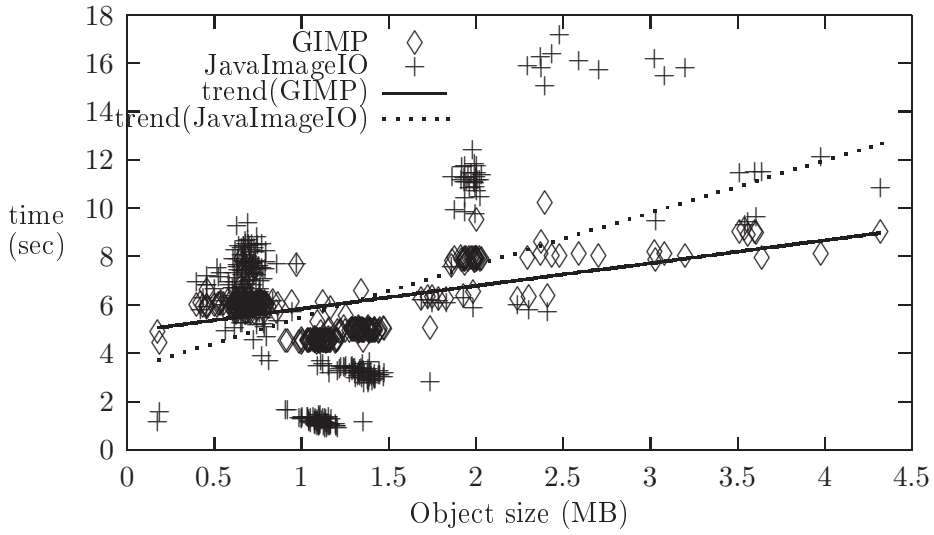


Figure 6.12: Processing speed of two migration components.

For Java systems, the actual memory within the virtual machine will be relevant, whereas in other cases, the virtual machine overhead has to be taken into account as well.

When a component is deployed as a service, a standard benchmark score is calculated for the server environment with the included sample data; furthermore, the monitoring engines report the average system load during service execution. This enables normalisation and comparison of a component across server instances.

Migration performance

Figure 6.12 shows the processing time of two migration components offered by the same service provider on 312 image files. Linear regression shows the general trend of the performance relation, with a root mean squared error of residuals of 1.01 for GIMP and 3.36 for JavaImageIO. The Java component is faster on smaller images but outperformed on larger files. It further is noted that the *conversion quality* offered by GIMP is certainly higher: Figure 6.13 shows a visualisation of a conversion error introduced by the simple Java program when converting an image with transparent background from GIF to JPG. Changed pixels are shown in grey in the figure and indicate that the transparency layer has been lost during migration. ImageMagick *compare* reports that 59.27% of the pixels are different in the migrated file, with an RMSE of 24034. In most cases, the information loss introduced by a faster component will be considered much more important than its speed, which

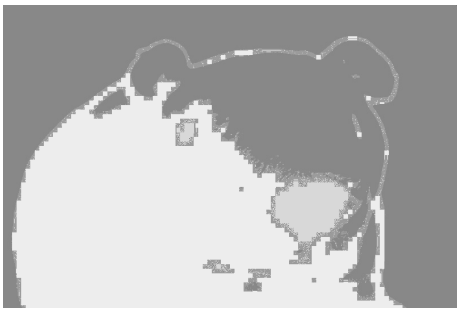
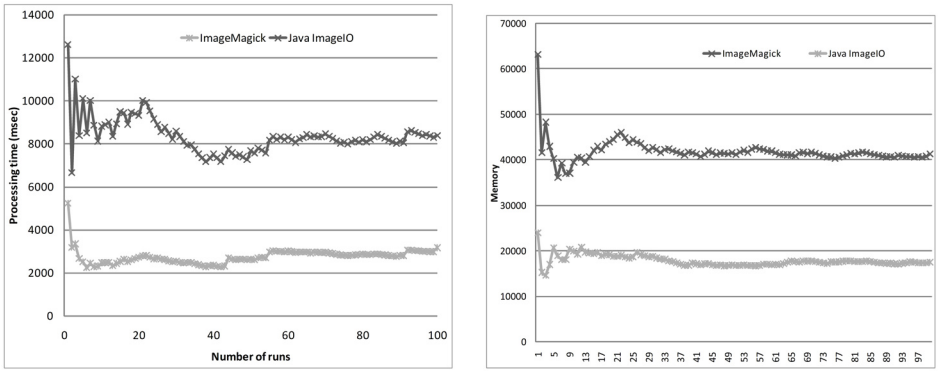


Figure 6.13: Visualisation of an examplary conversion error



(a) Processing time per MB (b) Memory usage per MB

Figure 6.14: Accumulated average performance data

in decision making will be reflected by the utility function and the weighting of criteria.

Accumulated experience

An important aspect of any QoS management system is the accumulation and dissemination of experience on service quality. The described framework automatically tracks and accumulates all numeric measurements and provides aggregated averages with every service response. Figure 6.14 shows how processing time and memory usage per MB quickly converge to a stable value during the initial bootstrapping sequence of service calls on benchmark content.

Tradeoff between QoS criteria

In service and component selection situations, sometimes a trade-off decision has to be made between conflicting quality attributes, such as cost versus speed or cost versus quality. When using the tool Free Lossless Audio Codec

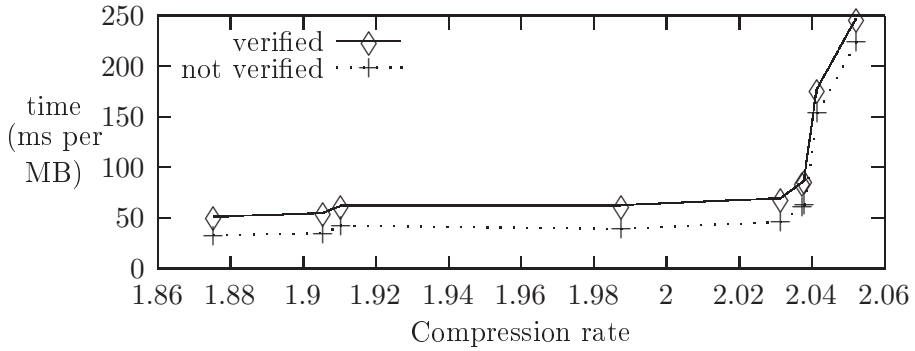


Figure 6.15: QoS trade-off between compression rate and performance.

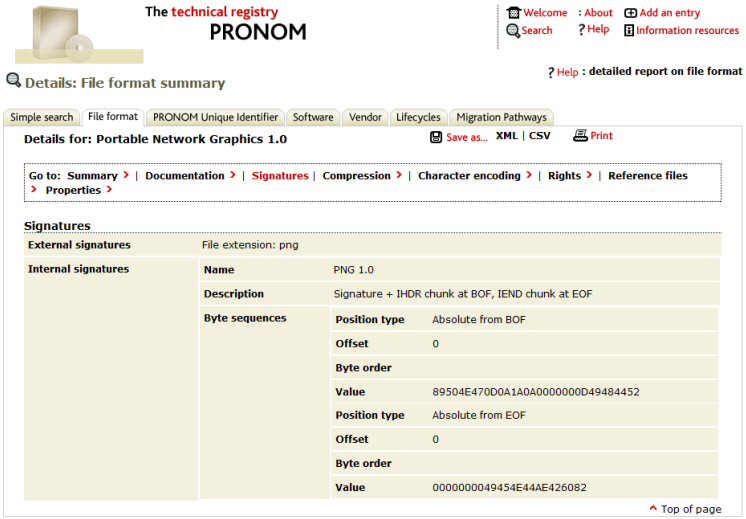
(FLAC)[14], several configurations are available for choosing between processing speed and achieved compression rate. In a scenario with massive amounts of audio data, compression rate can still imply a significant cost reduction and thus be a valuable tweak. However, this has to be balanced against the processing cost. Additionally, FLAC can verify the encoding process by including on-the-fly decoding and a comparison of the output to the original input. This provides integrated quality assurance and thus increased confidence at the cost of increased memory usage and lower speed.

Figure 6.15 projects the lossless compression rate achieved with nine different settings against used time. Each data point represents the average achieved rate and resource usage over the sample set from Table 6.3. It is apparent that the highest settings achieve very little additional compression while using excessive amounts of time. There is a consistent overhead incurred by the verification, but it does not appear problematic. Thus, in many cases, a medium compression/speed setting along with integrated verification will be a sensible choice, if compression is considered a viable option.

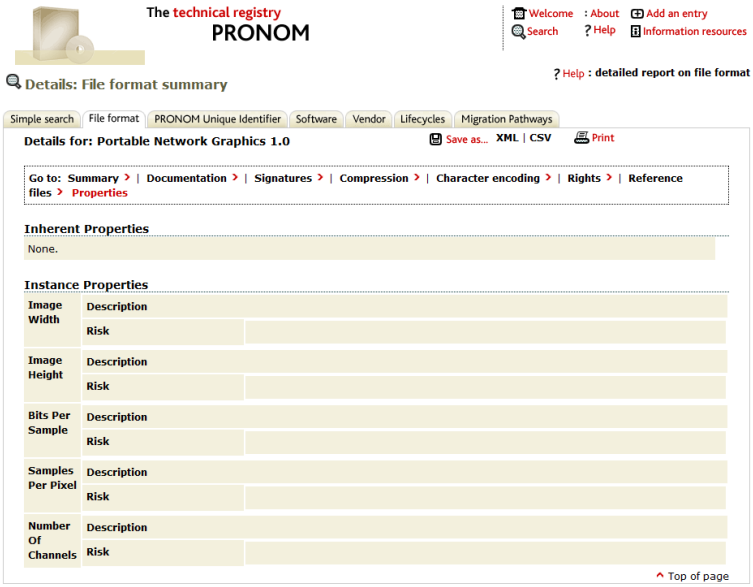
6.4.4 Summary

In this section, we have addressed the issue of measuring runtime characteristics of preservation action components in a flexible and extensible way. We described a framework for measuring dynamic runtime properties of components by monitoring them in a controlled environment. This generic architecture and framework for non-invasive provider-side service instrumentation provides quality-aware migration components. The resulting measurements are provided as metadata with the service execution and stored in the experiment results for automated evaluation.

[14] http://flac.sourceforge.net/



(a) PNG signatures in PRONOM



(b) PNG properties in PRONOM

Figure 6.16: PRONOM information about PNG 1.0

6.5 Accessing trusted information sources

While the last section has described dynamic measurements for process-oriented requirements, some of the criteria identified in the taxonomy lend themselves to being made available publicly at shared points of reference that can be trusted to provide accurate information. Especially criteria about file formats, which have since long been a focal point of analysis in the digital preservation community, are suitable to be described in publicly accessible registries. These are maintained by institutions with long-term commitment and substantial resources for evaluating certain aspects of formats.

Several points of information have been established in the past years to serve the interests of the digital preservation community. The most prominent examples are the PRONOM[15] Technical Registry maintained by the National Archives of the UK and the Global Digital Format Registry[16] (GDFR). PRONOM contains general information about formats and specific versions of formats. It provides descriptive information as well as persistent identifiers for versions of formats, and shows relationships between formats such as *PNG 1.0 is previous version of PNG 1.1*. It furthermore contains external and internal *signatures*, i.e. patterns that can be used by identification tools such as DROID to identify the format of files.

While PRONOM is owned and maintained by one single institution, the GDFR effort is geared toward shared governance and distributed data hosting. The recently established Unified Digital Format Registry[17] (UDFR) is a joint initiative begun in April 2009 to build a single shared formats registry.

The content of these registries so far is generally considered as trustworthy; however, specific information about file format properties and preservation tools is incomplete at best. For example, Figure 6.16 reveals that while PRONOM contains very specific description for identifying PNG formats, the level of detail about PNG properties is rather scarce. Furthermore, the current version 4 does not contain information about tools that can read certain formats. Most importantly, it includes only a fraction of the formats that are in use today.

Combining information sources to enhance the level of information available is thus clearly desirable. To this end, Tarrant presents the P2 registry[18] which uses Semantic Web technologies to combine the content of PRONOM, represented as RDF[19], with additional sources such as DBpedia[20] [THC09]. The P2 fact base currently contains about 44.000 RDF statements about file formats and preservation tools.

[15] `http://www.nationalarchives.gov.uk/PRONOM/Default.aspx`
[16] `http://www.gdfr.info/`
[17] `http://www.udfr.org/`
[18] `http://p2-registry.ecs.soton.ac.uk/`
[19] `http://www.w3.org/RDF/`
[20] `http://dbpedia.org/`

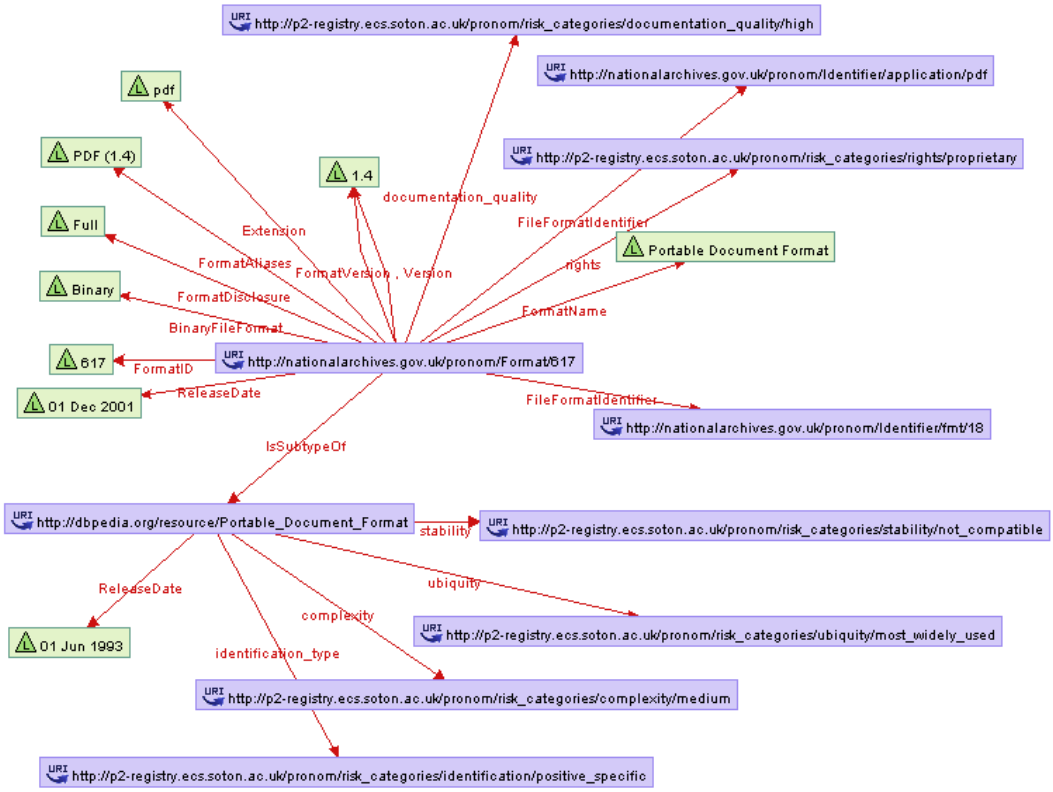


Figure 6.17: RDF graph showing some of the facts about PDF 1.4 contained in P2

Figure 6.17 shows a fragment of the RDF graph containg several facts about PDF version 1.4 as displayed in RDF Gravity[21]. PRONOM states among other facts that the format has been released on December 1, 2001 and that the *rights* are proprietary. It further assigns a Pronom Unique Identifier (PUID) of *fmt/18*. DBpedia does not contain specific information about this version of PDF. However, it contains a number of facts about the family of PDF formats, a few of which are shown in the lower part of the figure. Specifically, DBpedia contains tools that are able to view, render, convert, and create PDF files, and states that the format (family) was released on June 1, 1993. A large number of statements about tools able to read or write the format are not shown here. The P2 ontology connects facts from both sources and thus enables unified queries and reasoning over the entire graph [THC09].

We use the RDF facts contained in P2 and integrate these with our

[21] `http://semweb.salzburgresearch.at/apps/rdf-gravity/`

```
prefix pronom: <http://pronom.nationalarchives.gov.uk/#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?d WHERE {
  ?format pronom:FormatDisclosure ?d .
  ?format pronom:FileFormatIdentifier ?ident .
  ?ident pronom:IdentifierType "PUID" .
  ?ident pronom:Identifier $PUID$
}
```

Figure 6.18: SPARQL query for extracting the disclosure of a format

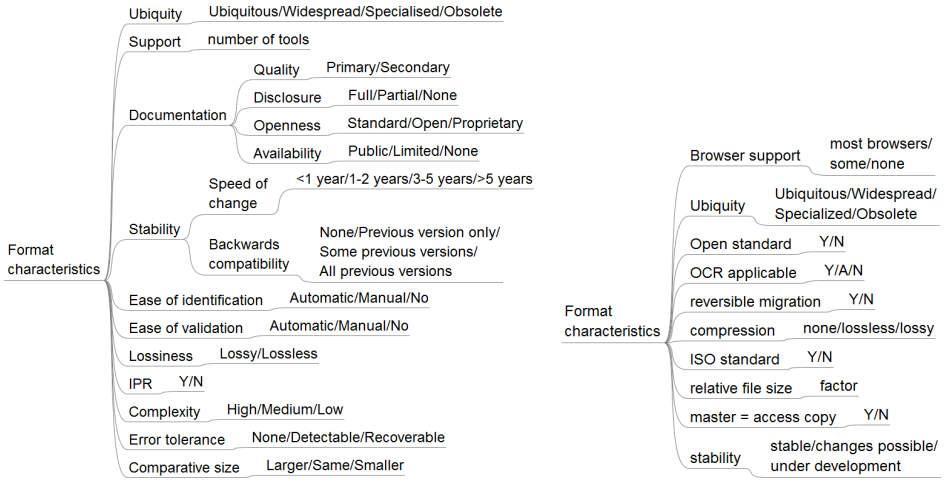
Property	**Scale**
Format disclosure	Full; Partial; None
Ubiquity	Most widely used; widely used; Occasional; Specialised; Deprecated; Obsolete
Documentation quality	High; Medium; Low
Rights	IPR protected; Open; Proprietary
Stability	Stable; Compatible; Not compatible; Unstable
Identification	Positive specific; Positive generic; Tentative; Unidentifiable
Complexity	Low; Medium; High
Number of viewers	Positive integer
Format age	Positive integer (years)
Newer version available	Yes; No

Table 6.4: Object format properties obtained from the P2 fact base

planning environment using a Jena triple store[22] and SPARQL[23] engine. The resulting Minimal Registry for the Extensible Evaluation of Formats (MiniREEF) is integrated in the planning tool through a query resolver. Figure 6.18 shows a basic query on the PRONOM fact base which upon provision of a PUID returns the disclosure of a format, which can be *Full, Partial*, or *None.* This is considered one of the basic risk factor of formats since a format with an entirely closed specification poses the risk that when the specification's owner ceases to exist, information about the format might be lost. Factors such as these have been the focus of thorough analysis [AF10, Flo08, GC04, LKR+00, Sta04, The08, Tod09]. Recommendations on which factors to include vary only slightly across literature; a lot of the recent work has been geared towards evaluating commonly used formats with respect to the criteria generally regarded as significant. These criteria correspond to the properties contained in P2, and include most of the factors shown in Figure 6.19 which shows two example criteria sets used in case studies. Several criteria appear in both trees; some, however, are specific to the institution and scenario. Table 6.4 lists some format properties that can be obtained

[22] `http://jena.sourceforge.net`

[23] `http://www.w3.org/TR/rdf-sparql-query/`



(a) Format criteria set A (b) Format criteria set B

Figure 6.19: Two example criteria sets for format evaluation

```
prefix pronom: <http://pronom.nationalarchives.gov.uk/#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT distinct ?swname WHERE {
   ?sw ?link ?format .
   ?link rdf:type
    <http://p2-registry.ecs.soton.ac.uk/pronom/SoftwareLink/Open> .
   ?format pronom:FileFormatIdentifier ?ident .
   ?ident  pronom:Identifier $PUID$ .
   ?ident  pronom:IdentifierType "PUID" .
   ?sw pronom:SoftwareName  ?swname
}
```

Figure 6.20: SPARQL query for extracting the tools able to read a format

from P2. The evaluation of these criteria provides a risk assessment for the considered target formats.

P2 contains a risk calculation model, where the organisation's preferences are captured in a *risk profile* that models the sensitivity of the organisation to certain risk factors. The outcome is a numerical risk score alongside a summarising analysis of the factors [THC09]. In our approach, the utility function fulfils this role by transforming the measurement of each criterion according to the acceptance thresholds of an organisation. Together with the weighting of factors, it models the risk aversion curve of the institution. This, in turn, is influenced by the policy defined by the organisation.

An old format that is readable by a very high number of tools is probably very stable and more suitable than a new format with a small number of tools. However, an old format with newer successors, but a small num-

```
prefix pronom: <http://pronom.nationalarchives.gov.uk/#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix p2-additional:<http://p2-registry.ecs.soton.ac.uk/ontology/#>

SELECT ?d  WHERE {  +
  ?format p2-additional:ubiquity ?u .
  ?u rdfs:comment ?d .
  ?format pronom:IsSupertypeOf ?pronomformat .
  ?pronomformat pronom:FileFormatIdentifier ?ident .
  ?ident pronom:IdentifierType "PUID" .
  ?ident pronom:Identifier $PUID$
}
```

Figure 6.21: SPARQL query for extracting the ubiquity of a format

ber of associated tools might be in high risk of obsolescence. The degree of adoption of a format could ideally be expressed as a market share; however, market shares are rarely known exactly. A secondary indicator is the number of tools that are able to process the format – obviously, a high number of tools available to open a format indicates a high interest and wide-spread adoption. Figure 6.20 shows a unified query over the RDF graph returning all tools that are able to *open* PDF files. Note that this number is particularly volatile in reality, and registries that are manually maintained by certain organisations will not be able to capture dynamic changes quickly. An entirely different approach for estimating the degree of adoption of a file format could rely on a trend analysis based on web content, similar to the approach presented in [MG09]. Such an approach would be particularly well suited to establish an automated *watch* after decision making in order to monitor the environment for substantial changes and raise an alert when a particular format is becoming obsolete.

Some of the other factors provided in Figure 6.19 cannot be directly mapped onto one criterion in the fact base and require more complex queries. For example, *stability* is a combination of the age of a format with the number of successors and the frequency of updates.

A third type of query is shown in Figure 6.21: The ubiquity of a format is defined in the P2 ontology as an ordinal judgement referring to a PRONOM category that corresponds to the criterion defined in Figure 6.19(a). The ubiquity, however, is assigned to the supertype PDF, not the specific version.

It should be noted that the measurement scales defined in these criteria trees are not always consistent with the value ranges obtained from P2. This stems from the historical fact that they were defined earlier. While some obvious mapping can be achieved through ontologies, standardisation of these criteria values is highly desirable. The knowledge base of the planning tool provides value ranges for measurable properties to ensure not only

repeatability of measurements, but also comparability across organisations and experience building in the community.

6.6 Integration with the planning tool

To integrate and access the evaluation modules described in the previous sections in the planning tool, the knowledge base of Plato has been extended to store a growing number of *measurable properties*. These are identified by a measurement information that consists of

- a measurement domain, i.e. top level category,
- a property pathname which is unique within this category, and
- an optional metric to be applied on the base measure that is obtained by the property name.

Each property can thus be assigned a unique URI, stating its domain, a unique name, and optionally a metric. For example, the significant property *image width* is generally measured in pixel and will usually be required to be left unchanged. We can thus specify a property

```
outcome://object/image/imagewidth#equal
```

defining an *outcome object* criterion for images named *imagewidth* to be compared using the Boolean metric *equal*.

To obtain measures for each property, a number of *Evaluators* is registered in the knowledge base and associated with the properties that each Evaluator is able to measure. Leaf criteria in the objective tree can be mapped to such a measurable property. For each mapped criterion, the corresponding evaluator will be invoked automatically during the evaluation stage.

Different strategies can be employed for discovery and invocation of evaluators. One is to simply iterate through the criteria list, look up the corresponding evaluator for each criterion as identified by the measurable property definition, and invoke it on this criterion to provide an evaluation result. However, this does not prove very scalable in cases such as XCL evaluation and extractors working on structured information sources, where considerable overhead is involved in the extraction procedure, but many criteria can then be evaluated at once. Furthermore, we observe that some evaluators fail to measure a certain value for one object or action, but another evaluator might succeed. This leads to the approach of a chain of evaluators grouped according to their category in the taxonomy.

We thus start with the full set of leaves and a prioritised sequence of evaluators. These are invoked in a certain order according to priority; each

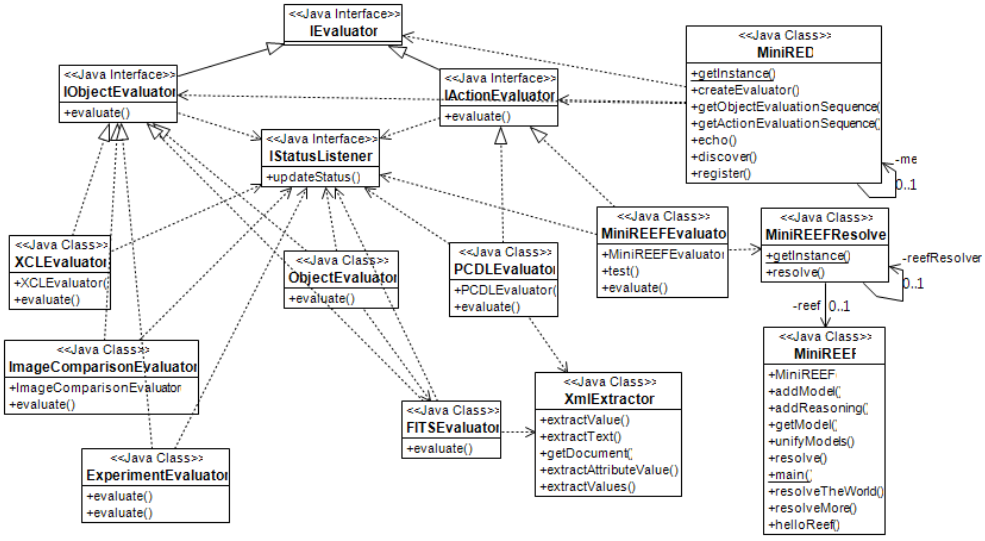


Figure 6.22: Currently deployed evaluators

successfully evaluated criterion is removed from the set of criteria to be evaluated. Each evaluation result includes provenance information documenting the measurement procedure. For example, in the case of significant properties comparison, the comparison result includes a documentation on both measured values and information about the way they have been obtained and compared.

Figure 6.22 shows the evaluation plugins currently deployed in the planning tool. There are two main categories: `IActionEvaluator` is the basic interface for evaluation of attributes that do not vary per object, while `IObjectEvaluator` is used for evaluating the outcomes of experiments on specific objects. A status listener interface provides a feedback mechanism for longer-running evaluation processes. The Minimal Registry for Evaluator Discovery (`MiniRED`) shown on the upper right provides the evaluator discovery point. The following evaluators are implemented:

1. The `XCLEvaluator` integrates the XCL tools described in Section 2.6 in the way outlined in Section 6.2 and thus delivers measures about the loss of significant properties induced by a preservation action.

2. The `ImageComparisonEvaluator` extends this by integrating measures from ImageMagick *compare* and other tools, as outlined in Section 6.2.

3. The `FitsEvaluator` focuses on the integration of criteria extracted by FITS as described in Section 6.3. To this end, it relies on a generic `XMLExtractor`.

4. The latter is also used by the `PCDLEvaluator` which extracts component properties from XML descriptors corresponding to a Preservation Component Description Language (PCDL) defined in *MiniMEE*. Components contained in the *MiniMEE* registry are described using such a language.

5. The `ExperimentEvaluator` analyses experiment data to deliver empirical measures about process characteristics. This includes extraction of information deposited by the *MiniMEE* engines described in Section 6.4, but also further evaluation of experimental data coming from other sources, such as log file analysis.

6. The `MiniREEFEvaluator` encapsulates the RDF triplestore containing the P2 fact base as represented by `MiniREEF`. It uses the `MiniREEFResolver` for executing stored queries such as those discussed in Section 6.5.

Table 6.5 lists some of the measurable properties currently stored in the knowledge base[24] and provides evaluators and sample results. Note that while the results may be stating only *'Yes'* or *'24'*, in fact each value contains extensive documentation about the measurement procedure. For example, for our by now well-known criterion *image width unchanged*, the evaluator will provide both measures together with information on the measurement source (such as FITS using JHOVE characterisation results). In the case of querying the number of tools available to read a format, the documentation also contains a complete list of tool names obtained from `MiniREEF`.

The evaluation framework is completely extensible and can be easily complemented with modules measuring different input sources, as long as they implement the basic `IEvaluator` interface. Envisioned extensions for the near future include the in-depth analysis of metadata schemes as well as an increased coverage of criteria extracted by FITS. Longer-term ideas include radically different approaches such as the integration of crowd-sourced evaluation frameworks similar to *reCAPTCHA*[25], as we will outline in Chapter 7.

The planning tool provides an expert interface to specify measurable properties and connect them to criteria trees and fragments. For example, we can define a tree fragment specifying significant properties of images, and tree fragments for format evaluation and typical process characteristics. We can further create reusable template trees for different scenarios of decision making about image preservation. Both fragments and complete trees are then accessible in the planning process.

[24] Two special properties extracted by XCL may require some additional context. The Levenshtein distance, also called edit distance, measures the amount of difference between two sequences [Bau88, Doy05]. PANOSE is a typeface matching system designed to classify

URI	Description	Evaluator	Sample value
action://runtime/activityLogging/format	Format of activity log output	Experiment	XML
action://runtime/activityLogging/amount	Size of activity log output	Experiment	1422 characters
action://runtime/performance/ time/perSample	CPU time used per sample	Experiment	877 msec
action://runtime/performance/time/perMB	CPU time used per MB	Experiment	348 msec
action://runtime/performance/time/ averageMemoryPerMB	Average memory load per MB	Experiment	17,4 MB
action://runtime/performance/time/peakMemory	Peak memory load of the migration process	Experiment	1824 MB
action://runtime/performance/ throughput/MBperSecond	Measured throughput of a component	Experiment	3,87 MB/second
outcome://format/documentation/quality	Documentation quality of a format	MiniREEF	Low
outcome://format/adoption/numberOfTools/Open	Number of tools than can open the format	MiniREEF	24
outcome://format/ubiquity	Degree of format adoption	MiniREEF	Widespread
outcome://format/IPR#exist	Are there any known IPR issues?	MiniREEF	Yes
outcome://object/format/conforms	Does the actual format conform to the declaration?	Object	No
outcome://object/relativeFileSize	Relative file size of results (factor)	Object	0.79
outcome://object/image/similarity#identical	Image similarity (AE other than 0)	Image Comparison	No
outcome://object/image/similarity#RMSE	Image similarity (RMSE)	Image Comparison	0.0
outcome://object/compression/scheme/lossless	Is compression lossless?	FITS	Yes
outcome://object/image/metadata#preserved	Are all (EXIF) metadata retained?	FITS	Yes
outcome://object/image/metadata/producer#equal	Are metadata on the producer retained?	FITS	Yes
outcome://object/image/metadata/ creationDate#equal	Are metadata on the creation date retained?	FITS	Yes
outcome://object/image/dimension/ aspectRatio#equal	Is the aspect ratio identical?	FITS	Yes
outcome://object/image/photometricInterpretation/ colorProfile/iccProfile#equal	Is the ICC Profile identical?	FITS	Yes
outcome://object/image/spatialMetrics/ ySamplingFrequency#equal	Is the vertical sampling frequency identical?	FITS	Yes
outcome://object/image/normData#equal	Are the normalised data identical?	XCL	Yes
outcome://object/document/normData#levenshtein	What is the edit distance of the normalised textual content?	XCL	44
outcome://object/document/ pageBackgroundColour#equal	Is the page background colour identical?	XCL	No
outcome://object/document/ documentLanguage#equal	Has the document language setting been preserved?	XCL	Yes
outcome://object/document/bbox#equal	Are the bounding boxes equal?	XCL	Yes
outcome://object/document/creationDate#equal	Has the document creation date been preserved?	XCL	Yes
outcome://object/document/fonts/panose#hamming	What is the average hamming distance of the PANOSE classification?	XCL	4

Table 6.5: Some measurable properties in the knowledge base

Nr	Type	Institution type	super-vised	Object format	Total	OO	OF	OE	AR	AS	AJ
1	documents	library	yes	PDF	44	27		2	1	10	4
2	documents	library	yes	PDF	33	19			4	8	2
3	documents	archive	yes	Word Perfect 5.x	38	35				1	2
4	documents	library	no	various	30	20		1	1	7	1
5	documents	research inst.	no	PDF	47	22	12	2		10	1
6	interactive	museum	yes	Game ROMS	81	58				22	1
7	interactive	research inst.	no	various	44	26				14	3
8	web archive	archive	yes	various	58	31	12	3		10	1
9	databases	archive	yes	MS Access	67	60	7				
10	images	library	yes	TIFF-5	24	8	6	1	3	3	3
11	images	library	yes	TIFF-6	33	18	10	2	1	1	1
12	images	library	yes	TIFF-6	40	10	12	1	3	10	4
13	images	library	yes	GIF	28	5	3	3	3	13	1
				Total	617	389	62	15	15	110	17
				Percentage	100%	63%	10%	2,4%	2,8%	17,8%	3,9%

Table 6.6: Distribution of criteria in case studies

6.7 Evaluation: Case studies revisited

We noted earlier that evidence is an essential precursor to trustworthiness, and that an entity's trustworthiness has to be evaluated in the realistic context of an action. Thorough documentation is needed to ensure reproducibility of evaluation experiments. One of the primary benefits of automated measurements is the degree of evidence and documentation that is produced along with the evaluation. We claim that under our stated conditions, these benefits can be achieved for a large fraction of the decision criteria. We evaluate this by quantitatively assessing the coverage of automated measurements with respect to criteria used in real-world decisions.

In this section, we thus evaluate the taxonomy presented at the beginning of this chapter and quantitatively analyse it to obtain indications of the coverage of automated measurements that can be achieved. We will first discuss the overall distribution of about 600 criteria that were collected from 13 case studies in Section 6.7.1. Section 6.7.2 will revisit the image case studies discussed in Chapter 5 to analyse the distribution of criteria and the quantitative improvement that has been achieved by the measurement framework.

6.7.1 Distribution of criteria

To answer the questions posed at the beginning of this chapter, we analyse a number of case studies that have been carried out during the last years with and without supervision and assistance from our side. Table 6.6 provides an overview of cases. All were searching for an optimal preservation component for preserving different types of images, documents, databases, web pages,

fonts according to their visual characteristics [Bau88, Doy05].

[25] `http://recaptcha.net/`

Category	Abbr.	Example	Data collection and measurements
Outcome object	OO	*Image pixelwise identical* (RMSE)	Measurements of input and output, measurements taken in controlled experimentation
Outcome format	OF	*Format is ISO standardised* (boolean)	Measurements of output, trusted external data sources
Outcome effect	OE	*Annual bitstream preservation costs* (€)	Measurements of output, trusted external data sources, models, partly manual calculation and validation, sharing
Action runtime	AR	*Throughput* (MB per ms)	Measurements taken in controlled experimentation
Action static	AS	*License costs per CPU* (€)	Trusted external data sources, manual evaluation and validation, sharing
Action judgement	AJ	*Configuration interface usability* (excellent, sufficient, poor)	Manual judgement, sharing

Table 6.7: Categories, examples and data collection methods

and interactive content. Most of the case studies were conducted in large repositories run by organisations such as national libraries, national archives, or large research foundations. Detailed discussions of several of these case studies can be found in [GBR08, KRB+09, BKG+09] and Chapter 5.

Two aspects about the circumstances of the studies are worth noting. Most of the studies were carried out with our assistance, but three of them were carried out independently without consultation, using the publicly available deployment of the planning tool. Furthermore, while most studies were evaluating components without a business-driven case of urgent action needs, three of the image preservation case studies (numbers 10-12 in Table 6.6) were actually delivering productive business decisions.

The categories in Table 6.6 correspond to the taxonomy described in Section 6.1. For each case study in the list, we provide the type of institution taking the decision, the type of content in need of preservation actions, and the number of decision criteria falling into each category. The bottom row summarises the distribution of the criteria. Of the 617 criteria that had to be evaluated, all fall into one of the categories of the taxonomy. 63% describe the significant properties of objects, while another 10% refer to desired characteristics of formats resulting from the application of components. Of the requirements on the components, their static properties constitute about 18%, while measurable runtime behaviour accounts for 2,8% of the criteria. This leaves 3,9% of criteria that fall into the categories *judgement of actions* and 2,4% that refer to general effects of outcomes, some of which have to be evaluated and calculated in a manual way.

Table 6.7 summarises the taxonomy's categories, maps abbreviations of Table 6.6 to the corresponding terms, and provides examples as well as the information sources needed for evaluation.

Some observations can be drawn from the statistics shown in Table 6.6. Some case studies have not defined any criteria in some of the categories. For example, several studies did not specify runtime action criteria; and some did not include any outcome effects. Two case studies that primarily evaluated emulation approaches for games (without ruling out migration) did not define criteria related to the object formats. In particular, the earlier case studies did not define format criteria. However, these are meanwhile usually included as essential risk factors.

The database study did not include any criteria related to the action. The reason was that the archive owns a substantial IT infrastructure and know-how and did not see the process as constraining the decisions. If an action would be expensive, take long, or be tedious to apply would not have influenced the decision, which they purely based on authenticity considerations and risk assessment. This is admittedly a rare case.

Considering the long-term development of preferences, it seems wise to still include these criteria in the decision tree with very low importance weights, if just to clarify explicitly that they had been considered, but not deemed important enough to include in the decision factors. Doing this would enable constant monitoring of preferences in the future to detect changes in the organisation´s priorities that have an impact on preferred actions. For example, a change in scalability demands may eventually require an attention to the scalability of components. This would also more strongly address requirements for trustworthiness that require an institution to be explicit about the factors that contribute to decisions and processes, and would provide traceable evidence.

6.7.2 Image case studies revisited

Compared to the overall distribution in Table 6.6, Figure 6.23 shows quite a different picture. It contains the distribution of criteria in the four recent image case studies discussed in Section 5.4 (Table 6.6, Studies 10-13). The distribution is significantly shifted compared to the overall averages and appears more balanced. While it is clear that the significant properties of images can be described with far fewer criteria than the properties of complex objects such as databases or even documents, the coverage of distinct categories of the taxonomy is evident.

Figure 6.24 shows a requirements tree derived from these image case studies. A ticker marks all criteria that are currently measured automatically. It shows that most of the *dynamic* properties are automated; what remains for manual judgement does not normally have to rely on in-depth studies of the objects or the dynamic behaviour of actions at processing time and can thus be evaluated rather quickly. Some criteria have been merged and/or reformulated in this tree for demonstration purposes. For example, format criteria like the fragments shown in Figure 6.19 and runtime characteristics

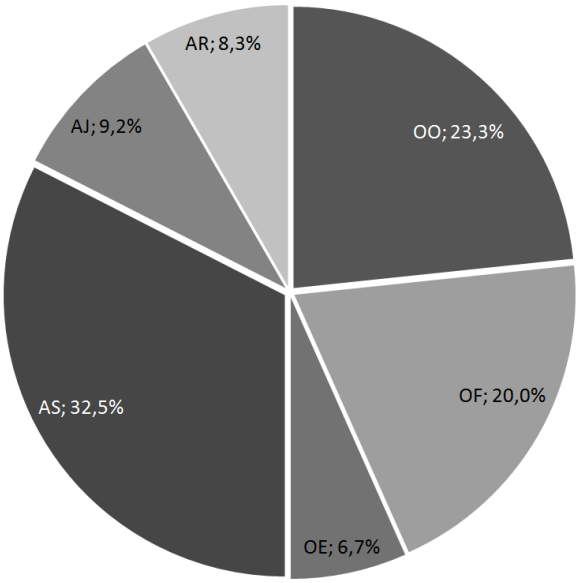


Figure 6.23: Distribution of criteria in image case studies

of the components have been homogenised compared to the original specifications. The criterion *ease of Planets IF integration* used in one study, which was requiring the tools to be easy to integrate in the Planets Interoperability Framework, was merged into a generic criterion *Compatibility with server environment*. Some criteria defined in the *process* branches of case studies were moved to the *component* branch because they are describing features of the components such as runtime behaviour.

On the other hand, specifics of each institution have been included in the tree shown in Figure 6.24, which thus represents a template from which we can select criteria in a given situation. For example, the criterion *Master can be used as access copy* was identified as relevant in one of the case studies, but can be of interest in others.

This reflects the converging knowledge about measurability and requirements. Measurable properties represent observable phenomena of interest in an objective and reusable way. Modelling the actual diversities of organisations and preferences is achieved by representing the differences through criteria selection, measurements, utility functions, and importance factors.

6.7.3 Coverage of measurements

Analysing the criteria in Figure 6.24, we see that the coverage of measurements differs significantly according to the high-level branches of the tree, which roughly correspond to the taxonomy categories. The overall coverage

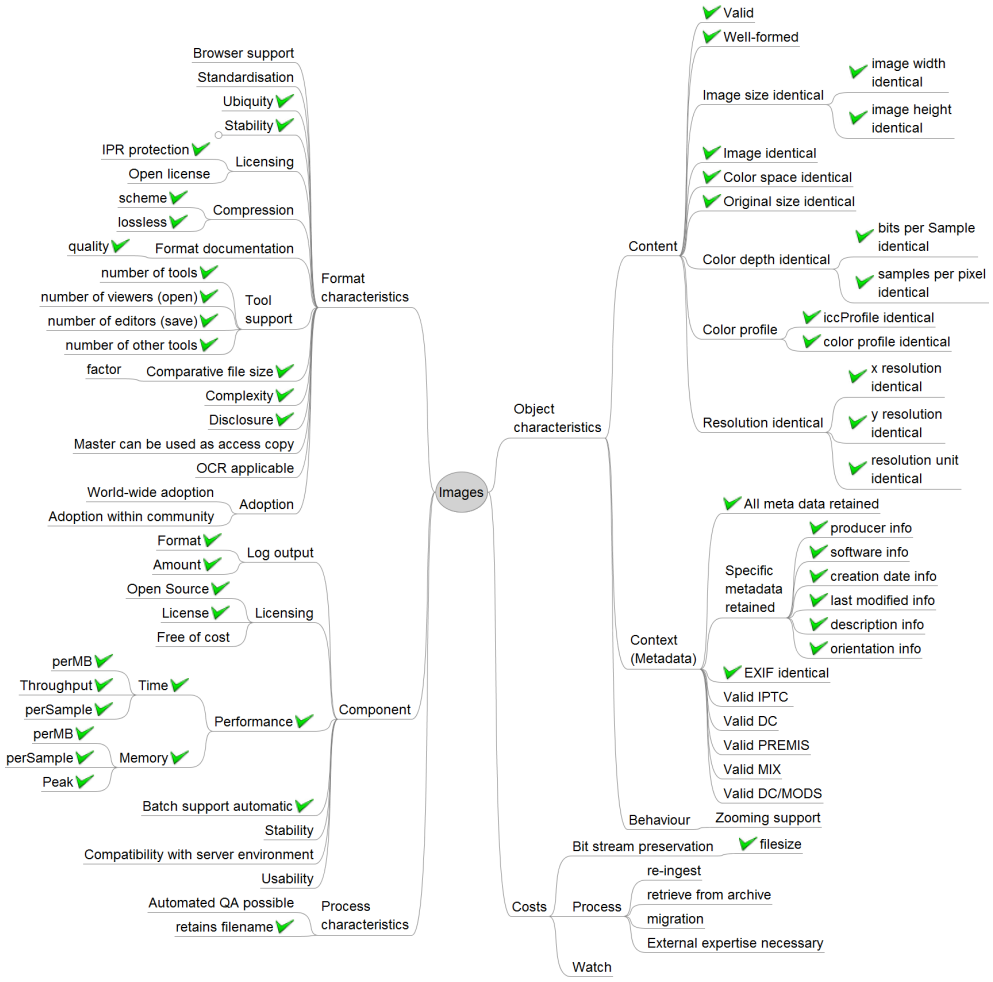


Figure 6.24: Image case studies: Automated requirements

of 67,6% (48 out of 71) of criteria is composed of coverage ranges between 16,7% and 100%. There is a full coverage of content criteria and a 61,5% coverage of context and metadata criteria. (It should be noted that the metadata criteria not covered at the moment can be easily included using the mechanisms described in Section 6.3 in the same way as they are used for measuring the already covered aspects.)

Costs naturally vary most, according to the costing structures of each institution, as discussed in Section 5.4. The only recurring property that has a direct influence and can be measured is the file size that influences bitstream preservation costs.

For component criteria, the runtime behaviour is fully covered, and so are most of the static properties in principle. However, the coverage that is achievable on these static criteria naturally depends on the availability of

the information to the extractors. Most of the criteria are described in the deployment descriptors of MiniMEE components, but MiniMEE contains only very few components. The other discussed registries, however, often do not provide this information.

A similar picture presents itself in terms of the formats: 13 of the 20 criteria are in principle covered, but this again depends on the completeness of the property specifications, i.e. the RDF graph in the P2 fact base. It would be possible to quantify world-wide adoption and, to a degree, even adoption within a certain user community, by monitoring trends on the web; however, this is not covered at the moment.

Considering more general cases than image preservation, the picture is of course less positive. The measurable aspects of components and formats comprise roughly 20% of the used criteria. However, we do currently not have mechanisms for measuring the behaviour of emulation environments in a scalable and generic way. The coverage of measurements for object characteristics of interactive content such as art and games is negligible at the moment, and similarly, there is no quality assurance accessible for comparing significant properties of databases. These comprise the majority of criteria and are therefore the key challenge to overcome. However, applying the principles discussed in Chapter 6 and the framework in this thesis, it will be possible (and necessary) to improve the coverage for complex object types. Chapter 7 will discuss future challenges and potential approaches.

6.8 Summary

In Chapter 5, we identified a number of key challenges to be addressed, based on real-world application experience.

- The definition of requirements and measurable criteria is technically challenging and complex. There is a substantial variation in the definition of significant properties, of performance characteristics, and of measurable properties in general. This also leads to a lack of comparability of results across case studies.

- The evaluation of the criteria is often unclear, and planners reported having difficulties in carrying out the evaluation procedure. The complexity of evaluating criteria such as the ones discussed in the case studies by studying the properties of the objects as extracted by characterisation tools and trying to figure out their meaning is overwhelming for many decision makers.

- The possibility to model organisational preferences and utilities is essential, but the objective *criteria* shold be standardised, reusable, uniquely identified, and selected from catalogues. Correspondingly, the measurements need to be clearly defined, repeatable, and reproducible.

This chapter presented an analysis of decision factors in preservation planning and showed how to improve the coverage of automated measurements through controlled experimentation and the systematic usage of external information sources. We showed that in principle, a majority of the criteria can be evaluated automatically. This not only reduces the effort needed to evaluate components, but also supports trust in the decisions because extensive evidence is produced in a repeatable and reproducible way and documented along with the decision in a standardised and comparable form.

For most of the criteria, there exist either trusted information sources or known means to automatically collect measurements to evaluate them. While for the multitude of formats and object types prevailing today, coverage of the significant properties measurements is still comparably low, the improvement in the image cases demonstrate that the coverage of decision factors can be significantly improved in practice.

However, there are several limitations that still need to be addressed, such as the modelling and handling of uncertainty, the measurement of significant properties for diverse and complex objects, automated monitoring, and the completion of the monitoring cycle to produce continously evolving and optimising plans. We will discuss these limitations of the current state of art in detail in the next and final chapter. This gap analysis will point out directions and next steps for future work that will address the identified challenges.

Chapter 7

Achievements, Generalisations and Limitations

7.1 Bringing it all together

7.1.1 The Challenges

While traditional non-electronic objects have to be saved from gradually fading away, the life curve of digital objects usually is cut off sharply: Incompatible environments or the inability to recognise the format an object is kept in will often mean that the object is lost entirely. For content which was born digitally, this loss is irrecoverable. To ensure the longevity of the ever-increasing amounts of digital information, a lot of which is comprising the cultural heritage of our times, is proving a continuing challenge. This thesis has focused on one of the key questions in digital preservation: The evaluation of potential actions to take to keep digital content alive, and the definition of trustworthy preservation plans. We presented an approach and tool to support the systematic planning of preservation actions through evidence-based, repeatable decision making and the thorough definition of well-documented preservation plans.

We commenced with a thorough analysis of the state of art in a number of relevant areas that come together in preservation planning. We analysed the OAIS Reference Model and its view on preservation planning, and we discussed criteria for trustworthiness in digital repositories as defined by catalogues such as TRAC and nestor. We observed that evidence is an essential precursor to trustworthiness.

We described an early case study on evaluating potential preservation strategies and analysed the challenges encountered. Evaluating potential actions such as migration and emulation components is difficult on a number of dimensions. Quality varies across tools; properties vary across content; usage varies across user communities; requirements vary across scenarios; risk

tolerances vary across content collections; preferences, costs, and constraints vary across organisations and environments. Finally, all of these factors are subject to constant shifts that have to be detected and handled.

We showed that the problem is a domain-specific instance of component selection and can be correspondingly reformulated and modelled. Domain specific adaptations are necessary and beneficial for our application scenario, but the component selection framework is applicable to other scenarios, as we discuss in Section 7.2.2.

A core technical challenge of digital preservation is ensuring the authenticity of digital content across different representations and renderings, i.e. verify that the application of preservation actions did not lead to a loss of significant properties, at least not to an intolerable extent. We described potential methods to ensure that this quality assurance can work effectively in a scalable way, and outlined the current state of the art.

Based on these observations and tools, we proceeded to build a framework and tool for trustworthy preservation planning in the subsequent chapters.

7.1.2 Preservation planning

Chapter 3 presented a systematic framework for preservation planning. We started by defining the necessary elements of a preservation plan and its structure.

> A preservation plan defines a series of preservation actions to be taken by a responsible institution due to an identified risk for a given set of digital objects or records (called collection). The Preservation Plan takes into account the preservation policies, legal obligations, organisational and technical constraints, user requirements and preservation goals and describes the preservation context, the evaluated preservation strategies and the resulting decision for one strategy, including the reasoning for the decision. It also specifies a series of steps or actions (called *preservation action plan*) along with responsibilities and rules and conditions for execution on the collection. Provided that the actions and their deployment as well as the technical environment allow it, this action plan is an executable workflow definition.[BKG+09]

The core part of our method for creating such plans is a component evaluation and selection procedure that relies on a variation of utility analysis to support this multi-objective decision making process. Our evidence-based approach to component evaluation can improve repeatability and reproducibility of component selection under the following conditions: (1) Functional homogeneity of candidate components and (2) High number of components and selection problem instances.

Its implementation in preservation planning results in five phases:

1. Define requirements,
2. Evaluate alternatives,
3. Analyse results,
4. Build preservation plan, and
5. Monitor requirements, quality of service, and the environment.

An analysis of the approach with respect to criteria for trustworthy repositories evaluated the contribution of the method towards building trust in a respository's operational planning.

7.1.3 Tool support

In Chapter 4, we presented the planning tool Plato, which incorporates a web frontend to a service-oriented architecture for evaluating preservation actions. The planning tool implements our preservation planning method and integrates registries and services for preservation action and characterisation. The tool enables preservation planners to create well-defined preservation plans. It provides substantial automation and guidance and documents all decisions made in the planning process. It furthermore provides a sophisticated web-based interface for guiding the planner through the process. We demonstrated the integration of action and characterisation components from a variety of sources through a flexible integration architecture. We described how the tool supports decision making and outlined the significant level of uptake it is experiencing world-wide in the digital preservation community.

7.1.4 Application

In Chapter 5, we discussed a number of case studies creating preservation plans for images, interactive content such as electronic art and video games, and databases. We specifically focused on a series of comparable case studies that resulted in real-world business decisions about situation-optimal strategies for preserving large image collections. We discussed why the evaluation of four cases resulted in three different recommendations. We further presented a number of common misperceptions and lessons learned through this real-world application experience. A critical assessment of the shortcomings and challenges of the approach led to the identification of the following critical gaps:

- The definition of requirements and measurable criteria is technically challenging and complex. There is a substantial variation in the definition of measurable properties, leading to a lack of transparency and comparability.

- The evaluation of criteria is often unclear, complex, and difficult in practice.
- The possibility to model organisational preferences and utilities is essential, but the objective *criteria* shold be standardised and reusable, and the measurements need to be repeatable and reproducible.

7.1.5 Improvements

To address the identified challenges, Chapter 6 presented an analysis of decision factors in preservation planning and described the construction of a taxonomy of decision criteria. We then showed how to improve the evaluation by using automated measurements obtained through controlled experimentation and systematic usage of external information sources. We described an extensible evaluation framework and demonstrated measurements along the following aspects:

- Comparison of object characteristics to determine the degree of loss induced by a preservation action,
- Extraction of information from structured data sources,
- Adding quality-awareness to migration components through a non-invasive monitoring framework, and
- Integration of trusted information sources.

We claimed that under our given conditions, the benefits of automated measurements can be achieved for a large fraction of the decision criteria. We evaluated this by quantitatively assessing the coverage of automated measurements with respect to criteria used in real-world decisions. We thus discussed the distribution of over 600 decision criteria, used in 13 real-world case studies, across the categories of the taxonomy. The analysis showed that in principle, a majority of the criteria can be evaluated automatically.

This not only reduces the effort needed to evaluate components, but also supports trust in the decisions because extensive evidence is produced in a repeatable and reproducible way and documented along with the decision in a standardised and comparable form.

7.1.6 Research questions revisited

In Chapter 1, we presented a number of research questions to be addressed in this theses. Particularly, these were:

RQ1: **How can we select the optimal preservation action for a given setting?**

Challenges to be addressed included decision space constraints, influence factors and preference structures, and the issue of modelling competing objectives and requirements to evaluate software components. These questions were addressed primarily in Chapter 3, where we presented a systematic component selection and preservation planning framework.

RQ2: **How can we ensure trustworthy preservation planning?**
Aspects to be considered included requirements on trust in repositories, necessary documentation, aspects of planning, as well as reliability and repeatability.

In Section 2.3, we presented an overview of criteria catalogues and assessment approaches for trustworthy repositories, and derived requirements on trust that are relevant for preservation planning. The core part of Chapter 3 defined the structure of a plan and presented a repeatable evidence-based evaluation process for creating such plans. Section 3.6 discussed the contribution of our method towards building trust in a respository's operational planning.

RQ3: **How can we ensure that decision processes scale up?**
Scalable planning requires substantial automation in terms of decision making, monitoring, and measurement. We identified a need for automation in decision making, integration of continuous monitoring, and thus particularly the question of measurement: What needs to be measured, and how can we measure it?

Chapter 4 presented the planning tool Plato that provides substantial automation in the decision process. A critical assessment of the challenges of our approach based on a number of real-world case studies in Chapter 5 led to the construction of a taxonomy of criteria and a measurement framework designed to support scalable decision making in Chapter 6, including a quantitative coverage analysis.

7.1.7 Limitations

Several limitations of the current state of the art need to be addressed in the future. On the one hand, the implementation of the evaluation framework currently relies on being *extensible* rather than completely dynamic and pluggable. A more flexible and scalable approach is needed for large-scale automation.

On the other hand, the raw number of criteria alone is not sufficient to estimate the effort needed for measurement, and thus does not fully reflect the state of progress achieved and the issues still to be addressed. Furthermore, it does even less reflect the actual quantitative influence each criterion

has on the decision making process. Corresponding to the utility aggregation and ranking described in Chapter 3, this is felt on two levels:

1. A criterion with a utility function that has an output range including zero has the potential to reject components that fall outside the acceptance range.

2. Depending on the relative weights in the tree, the depth of the tree hierarchy and the total number of criteria used for evaluation, a criterion will have a varying overall influence on the ranking at the root level.

Apart from this quantitative impact prioritisation, we have to consider the operational costs of obtaining the measurement automatically and the effort needed for manual evaluation. Based on these factors, we plan to conduct a quantitative assessment of decision criteria, their measurement costs, and their quantitative impact. The outcome of this analysis will provide directions to optimise and automate decision-making, watch, and policy definitions at large scales and enable smaller organisations to make decisions at a lower entry barrier by focusing on the essential factors (trade-off). It will furthermore increase the focus and impact of research in characterisation and quality assurance by creating a roadmap that prioritises the aspects that are most urgent and have the strongest impact.

More advanced issues include the modelling and handling of uncertainty, the measurement of significant properties for diverse and complex objects, the incorporation of quality-awareness in emulation, the implementation of automated monitoring, and the completion of the monitoring cycle to produce continously evolving and optimising plans. Section 7.3 will discuss these challenges in detail and point out opportunities and intentions for future work.

7.2 Wider applicability

While the core focus of this thesis is on digital preservation and preservation planning, several aspects of the presented work have wider significance. Specifically, the approach to quality-aware service provisioning and the improvements that controlled experimentation and automated measurements bring to component selection under certain conditions have been contributed to the areas of web engineering, software engineering, and requirements engineering [BKK+09a, BR10, BR09]. Furthermore, the planning tool can be generalised to support multi-objective decision making in scenarios other than logical preservation planning.

7.2.1 Quality-aware service provision

As described in Chapter 4, we rely on a service-oriented architecture for component discovery and invocation. Service-oriented computing as means

of arranging autonomous application components into loosely coupled networked services has become one of the primary computing paradigms of our decade. Web services as the leading technology in this field are widely used in increasingly distributed systems. Their flexibility and agility enable the integration of heteregoneous systems across platforms through interoperable standards. However, the thus-created networks of dependencies also exhibit challenging problems of interdependency management. Some of the issues arising are service discovery and selection, the question of service quality and trustworthiness of service providers, and the problem of measuring quality-of-service (QoS) attributes and using them as means for guiding the selection of the optimal service for consumption at a given time and situation. These aspects apply equally to our scenario: Evaluation strongly depends on objective measurements of the components under consideration.

Measuring quality attributes of web services is inherently difficult due to the very virtues of service-oriented architectures: The late binding and flexible integration ideals ask for very loose coupling, which often implies that little is known about the actual quality of services and even less about the confidence that can be put into published service metadata, particularly QoS information. Ongoing monitoring of these quality attributes is a key enabler of service level agreements and a prerequisite for building confidence and trust in services.

Different aspects of performance measurement and benchmarking of web services have been analysed. However, most approaches do not provide concrete ways of measuring performance of services in a specific architecture. Detailed performance measurement of web services is particularly important for obtaining quality attributes that can be used for selection and composition, and for discovering bottlenecks to enable optimisation of composite service processes.

While client-side measurement is certainly a valuable tool and necessary to take into account the complete aspects of web service execution, it is not able to get down to the details and potential bottlenecks that might be negotiable or changeable, and thus benefits from additional server-side instrumentation. In large-scale systems, measuring the performance of components in detail can be crucial.

In Section 6.4, we described an extensible monitoring framework for enriching web services with QoS information. Quality measurements are transparently obtained through a flexible architecture of non-invasive monitoring engines. We demonstrated the performance monitoring of different categories of components and discussed different techniques and the results they yield.

While the resulting provider-side instrumentation of services with quality information is not intended to replace existing QoS schemes, middleware solutions and requester-feedback mechanisms, it is a valuable complementary addition that enhances the level of QoS information available and allows

verification of detailed performance-related quality criteria. Moreover, this provider-side measurement allows service requesters to optimise access patterns and enables service providers to introduce dynamic fine-granular policies such as performance-dependent costing. The resulting design principles for scalable quality-aware components are specifically being taken forward in follow-up projects designing large-scale preservation environments, but are applicable to broader areas.

7.2.2 Improved component selection

Component selection and evaluation is a continuous problem space ranging from Component Based System Development to web services and other dynamic scenarios, as discussed in Section 2.5. Our component selection method has in its genericity been outlined in Section 3.3 and is described in detail in [BR10]. It is designed for settings sharing the following characteristics:

1. *Homogeneous functionality.* – Since the functionality of components is homogeneous and well-defined, and competing tools provide essentially the same functionality, it is feasible to create dedicated evaluation modules.

2. *Continuous evaluation and monitoring.* – The selection process has to be repeated regularly, potentially leading to a reconfiguration or replacement of components. Hence, there is a need for continuous evaluation and monitoring.

3. *Transparent and auditable decisions.* – Since the requirement of trust in software components and services is critical, decision making and component selection procedures need to be fully transparent and reproducible to provide sufficient levels of accountability. A thorough and objective documentation about the information that was available at the time of decision making is thus of vital importance.

The presented evidence-based approach to component evaluation can improve repeatability and reproducibility of component selection under the following conditions: (1) Functional homogeneity of candidate components and (2) High number of components and selection problem instances. In machine translation, components have similarly focused and well-defined functionality, and techniques for automated evaluation of translation quality are being developed [PRWZ02, Dod02, LRL05]. Again, thorough evaluation and continuous monitoring of a translation component is required to cope with e.g. topic drift in the source documents. Similar requirements can be found in numerous application domains such as compression tools or sort and search in high-dimensional index structures.

The presented taxonomy is geared at digital preservation; however, it is generally applicable to domains sharing these characteristics. In machine translation, for example, a taxonomy would likely be very similar apart from the *format* category, which still may be present, comprising structural characteristics of the resulting translation following a standard representation. Object criteria may further include the presence of confidence values or alternative translation candidates for certain terms.

7.2.3 Tool support for multi-objective decision making

The planning tool Plato has been developed specifically for preservation planning, and its obvious application has thus been the evaluation of preservation action components. However, we have used it successfully to evaluate bitstream preservation strategies in a study [BR07] conducted for the Austrian Chamber of Commerce[1], and it is currently being used for another evaluation of bitstream preservation strategies in large institutions.

The domain-specific information that is being documented in the decision process, the terminology used, and the integrated modules for preservation action components are geared towards digital preservation. Yet, the underlying concepts, the workflow, the requirements specification and evaluation model, and the corresponding software design and modules are domain-independent and can certainly be useful in other scenarios. We aim to release a generalised decision support tool based on the current design and code of the planning tool.

7.3 The future of preservation (planning): Current challenges

While the previous sections have summarised the main aspects of this thesis, this section takes a look forward, based on the state of the art produced thereby and the corresponding insight into the shortcomings and upcoming challenges. We identify a number of challenges to be addressed in the near and medium-term future:

- Reduction of complexity,
- Improvement of measurement techniques,
- Handling of measurement reliability and uncertainty,
- Integration of planning in repositories,
- Continuous monitoring and impact assessment,

[1] `http://www.ifs.tuwien.ac.at/dp/fotostudie/`, in German

- Planning as a Service, and
- Scalable preservation planning.

7.3.1 Reduce complexity

Even with the improvements presented in this work, preservation planning is a complex procedure that at first may overwhelm decision makers. Plato provides considerable support and enables planners to reuse experience of others through a shared knowledge base. Still, the overall complexity of the problem implies that sophisticated tool support is needed to pro-actively guide decision makers and help them where possible in selecting information and making the right decision. To this end, recommendation modules are currently under investigation that shall operate on case-based reasoning concepts. This may include some of the following aspects.

- Automated selection of representative sample content based on large-scale in-depth collection profiling;
- Automated construction of criteria trees with a certain coverage of influence factors, based on formalised policy models that reflect environmental and organisational constraints;
- Automated construction of significant property trees based on a combination of templates and properties extracted from the sample objects;
- Automated construction of utility functions based on measured values, policies, and (aggregated) user feedback; and
- Automated suggestion of candidate components to include in the evaluation phase, based on shared experience bases and aggregated utility values.

Furthermore, the 14-step workflow for preservation plan definition is a high entry barrier for first-time planners, and an unnecessary overhead for users who only want to conduct quick experimentation. While other developments such as the Planets Testbed[AHJ+08] partly address these evaluation needs, they do not provide the sophisticated requirements specification and utility approach of Plato, the hierarchically structured visual analysis of strengths and weaknesses of each component, and the coverage of measurements presented here. We are thus aiming at supporting the fast-track evaluation of components using an accelerated highly-automated workflow that proceeds in three steps through the core three phase of requirements definition, experimentation, and analysis to produce an evaluation report.

7.3.2 Improve measurement techniques

While we provide an extensible framework for automated measurements and evaluation, actual automation in practice is to a large degree hindered by the lack of coverage provided by available measurement tools. The XCL languages still cover only a fraction of the content types used in practice; and tools such as FITS do not deliver in-depth measurements of complex objects such as databases and interactive content. Even worse, current emulators completely lack the ability to deliver quality measures about their accuracy in representing the original environment. To provide scalable evaluation for planning and operational application, we need to create quality-aware emulators that are able to contribute to the measurement of significant properties and authenticity, and we have to substantially increase the coverage of quality assurance for converted objects.

Future quality assurance tools further have to address the heterogeneity of content encountered in today's information landscape, particularly on the web. This includes standard internet content, emerging Web 2.0 and user created content such as the Social Web, but also the Deep Web, complex interactive objects, databases, and scientific data. In particular, digital longevity is presenting a real challenge in domains which have high volumes of heterogeneous and complex high-value content to preserve, such as manufacturing, finance, pharmaceutical companies, medicine, and e-Science [Man10].

Four specific approaches are particularly interesting and show promising potential.

- Benchmark content can be generated, instead of characterised.
- Crowd-sourcing can contribute to scalable quality evaluation.
- Quality assurance can be carried out on the perceptual level.
- Semantic open-world models can capture evolving knowledge.

Don't characterise: Generate

The development and improvement of current characterisation techniques is still very much hindered by a fundamental lack of standardised benchmarks. Annotated benchmark data are needed to support the objective comparison of new approaches and quantify the improvements over existing techniques. This lack of baselines is partly due to the fact that the creation of such benchmarks is extremely effort-intensive.

To ensure measurement reliability, the digital preservation domain has started to define criteria for benchmarking corpora and stratification of test data [NBL+07]. A baseline benchmark needs to rely on known ground truth. However, for many object types such as databases or electronic documents, this ground truth is never known beforehand, but instead has to be extracted

from the objects themselves. Since the variation in objects, their features, and formats and subformats is so high, there exists little safe ground on which to create a baseline for quantitative improvements.

The common approach so far has been to search for appropriate real-world collections, take a subset of these that is not protected by copyrights and other regulations, and then try to define the properties of that set of objects. But given the incompleteness of properties coverage and the lack of format coverage of current tools, these approaches have not yet led to reusable, well-specified benchmark data where the ground truth is solidly defined in a standardised way.

Instead of characteristing objects taken from real collections, a solid bootstrapping approach could rely on *generating* the test data from a fact base with desired properties. For example, consider that the starting point of a document is not the file representing it for example in the Word 97 format, but instead the document model as it is created in the text editor by the user. Correspondingly, the automated generation of test data for Office documents can rely on a domain-specific property definition language and a code generator that produces Macro code for Office software. This would support the creation of truly 'origin' documents – documents that are created in almost the same manner as if a human user would write them. It could lead to perfectly specified data sets and tackle the challenge of stratification since it would support the explicit and exact configuration of the desired variation of properties. The approach furthermore makes it easy to create representations in different formats supported by one program and analyse the exact variations in the produced bytestreams.

To this end, software engineering techniques for test data generation, coverage analysis and overlap detection should be employed to create benchmark data sets annotated with reliable ground truth information.

Leverage the wisdom of the crowds

Even with much more advanced characterisation and QA tools than those available today, there will probably always remain a certain gap between the features that tools can characterise and the features of brand new content as they are perceived by humans. To this end, crowd-sourcing frameworks could be leveraged, where fine-granular evaluation problems are posed to massive amounts of users, and the answers collected for aggregated analysis. Similarly, the incorporation of such user feedback mechanisms into the access frontends of repository systems would enable the collection and analysis of user perception feedback on large scales.

Conduct quality assurance on the perceptual level

Most current QA techniques operate on the level of file formats, trying to interpret the extracted properties from different formats and compare them to each other. As we have seen, this is faced with two major challenges:

1. The mapping of properties between formats is very often not homomorphic at all; even worse, there is often no clear way of creating such a mapping at all. Consider a Open Office XML document with a table converted into PDF. Inn OOXML the table is clearly identified, but a PDF extractor will have considerable difficulties in recognising it, depending on the way the PDF conversion tool has created the document.

2. The multitude of formats and their variations makes this kind of property extraction computationally intensive and error-prone.

Instead, it may be possible to achieve better results by evaluating characteristics on the perceptual level and analyse a trusted interpretation produced by a reliable, well-tested tool on a standardised reference platform. (Such reliability tests can be conducted using generated test data annotated with reliable ground truth.) For example, image analysis and OCR analysis can be conducted with open-source tools such as OCRopus[2] on electronic documents printed to TIFF using standardised reference renderings. Several approaches and tools for page analysis, page segmentation, and content understanding have been presented during the last years [EDG+02, GCMC02, CYWM03, CPIZ07, SGP09]. These can either be tested on generated test data or on realistic datasets for performance evaluation of document layout analysis such as the PRImA dataset[3]. Similarly, for complex audio recordings and multi-track audio content produced by sequencers, audio analysis may be applicable.

Capture evolving facts and knowledge

While automated QA for object transformation and emulators is very challenging, a considerable fraction of the criteria is seemingly easy to evaluate. The categories *Action Static* and *Outcome Format* together accounted for about 28.8% of the overall case study criteria and 52.5% of the criteria used in the image case studies.

The coverage of these criteria and the up-to-dateness of discovered facts is in reality still relatively low, because moderated registries such as PRONOM that are in use today are not dynamic enough to capture the evolving facts and the knowledge that is available, for example on the web. The implicit

[2] `http://code.google.com/p/ocropus/`
[3] `http://dataset.primaresearch.org/`

closed-world assumption in the design of these registries does not hold, since new facts will be discovered constantly. For example, a migration component may be defined as being *stable* in a static registry after initial testing. However, large-scale experiments might discover that the component is only stable for the commonly used input formats and in fact crashes on 12% of the objects in a certain different format.

Open-world models such as RDF and ontologies may be better suited to capture the inherently evolving nature of repositories, user communities, and technologies, and allow reasoning over known facts to produce derived knowledge.

7.3.3 Address measurement reliability and uncertainty

The above discussion about measurements reminds us of the inherent uncertainty that is imminent to the measurements that need to be taken. This uncertainty in measurements and judgements needs to be addressed on four levels:

1. Reliability of measurements,
2. Reliability of judgemenents,
3. Reliability of the utility functions, and
4. Handling uncertainty in the evaluation.

Annotated benchmark data is needed to provide the means for validating measurement accuracy of quality assurance tools such as the XCL languages, as discussed above. Furthermore, ecplicitly modelling the *confidence* we have in the reliability and precision of a measurement can inform sensitivity analysis and improve the robustness of decision making. The specificity of the measured entity and the precision of the measurement device may contribute to these confidence levels.

Consider the evaluation of the criterion *format adoption* for a subformat such as PDF 1.5. If the evaluation returns the adoption measure only for the PDF family, because the registry does not specify exact data for PDF 1.5, we may assume that there is an uncertainty in factor in this measurement, which will be related to the number of PDF subformats 'competing' which each other for market shares. Taking this uncertainty into account enables more robust decision making by including the potential variation of measures in the sensitivity analysis.

As noted before, there is still a certain percentage of criteria that can not be measured automatically and have to be judged by experts. This judgement naturally entails the risk of not being reproducible and exhibiting certain biases. The sensitivity analysis described in Section 3.3.4 partly

addresses this issue, but cannot provide guarantees. The usage of AHP may be beneficial for these criteria. We further aim at extending the evaluation platform to enable experience sharing and provision of aggregate statistics about such judgements. This sharing also benefits aggregated statistics of measurements taken in the controlled environment on different input data and can lead to a collaborative benchmarking platform.

Provided that a sufficient number of people have shared their judgements, the accumulated averages of these criteria may become *static* criteria, where the common converging judgement is used as evaluation value. As noted, this requires a shared participation and open-world model that is very different from the moderated content model currently prevailing in digital preservation registries such as PRONOM.

The currently used approach to sensitivity analysis provides a robustness measure of the decision maker's preference structure that takes into account the weightings of importance factors in the objective tree. However, it does not take into account the discussed measurement uncertainty, and does not handle the specifics of the scales that are used as input for the utility function. Since it only operates on the calculated utility, it fails to address the fundamental differences between ordinal and numerical scales: While uncertainty in ordinal values translates to a flip in the values that could be modelled by a randomised dice, the numerical (continuous and discrete) measurements show different variance. Taking these differences as well as a confidence value into account should provide better sensitivity analysis and more robust decisions.

7.3.4 Incorporate planning into repository operations

To become operationally usable in repositories, planning needs to be integrated in the regular repository software. This means that plans have to be created for specific identifiable sets of objects and carried out in the repository. Continuous monitoring needs to be addressed as well. We are currently working on the integration with several leading repository systems, closely collaborating with each of the development teams.

- EPrints[4] is one of the most widely used platforms for building institutional repositories.
- The Repository of Authentic Digital Objects[5] (RODA) is an open source digital repository based on Fedora. It has been developed in an initiative of the National Archive Institute of Portugal and was specifically designed for archives, with long-term preservation and authenticity as its primary objectives.

[4] `http://www.eprints.org/`
[5] `http://roda.di.uminho.pt/?locale=en#home`

- Mopseus[6] is a lightweight digital library service based on the Fedora system [AGCP09].
- eSciDoc[7] is an eResearch environment for scientific communities, comprising a Fedora-based repository with a set of additional services addressing eScience scenarios.

First steps towards basic integration with ePrints have been demonstrated [FTRK09]; full-fledged integration is planned for the future. That means that a repository shall trigger planning activities when risk assessment raises a corresponding alert, shall provide all necessary and available information to the planning environment, receive a corresponding plan addressing the risk, and then implement the plan. Following this line, we discover that for a fully operational environment to execute the plans, we need more than a discovery and invocation of the action component on all objects. The need for quality assurance and reporting requires the ability to construct complex executable plans and workflows integrating QA, metadata generation, reporting, and integrity checks. One promising technology to address this are workflow environments such as Taverna[8], an open source tool for designing and executing workflows.

7.3.5 Monitor continuous operations and detect changes

Following the definition of a preservation plan and its deployment in a repository, two aspects have to be monitored:

- Inward watch has to monitor the ongoing operations of the repository, which includes active deployed preservation plans, but also access patterns and ingest statistics;
- Outward watch has to monitor external influence factors in the repository's environment that may impact on preservation operations. This ranges from market shares of file formats to emerging browser technologies and shifts in designated communities.

Both general QoS attributes such as performance, throughput, resource usage etc., as well as specific QoS criteria such as accuracy need to be continuously monitored during operation to ensure that the deployed plan indeed keeps fulfilling the requirements as expected. Any deviation in QoS of the level measured during the experiments is an indication of either an incomplete evaluation procedure, or a change in the environment that needs to be

[6] http://www.dcu.gr/dcu/site/Projects/t_docpage?doc=/Documents/projects/mopseus

[7] http://www.escidoc.org/

[8] http://www.taverna.org.uk/

addressed, such as a sudden increase in data volume. Depending on the type and severity of the deviation, this may lead to a re-iteration of the planning procedure, where the original scenario is taken as a starting point and revised according to changes in the environment, technology, or the requirements.

Watch modules further have to extract information from the environment to monitor specific parameters that influence preferences and decisions. The most prominent example in this context is technology watch, where aspects such as the distribution of file formats are monitored and warnings can be raised when specific thresholds are exceeded.

These watch modules are an important basis for continuous monitoring and iterative planning. To this end, thresholds could be defined on various levels that trigger an alert when exceeded. These service level agreements could rely on the utility functions defined by the organisation to compute acceptable ranges for QoS attributes. Accumulated experience can inform the definition of monitoring conditions during the final integration stage of the planning workflow.

7.3.6 Scale down: Planning as a Service

While several large-scale approaches to digital preservation have been fairly successful, smaller institutions and individuals have not yet been able to take advantage of these methods and tools. Yet, a large amount of information, comprising an enormous value, is created and stored every day by private users and small organisations. This ranges from family photographs and videos to emails and other types of documents created in virtually every household and office. Small and medium enterprises face similar challenges concerning their core business documents. These objects need to be preserved through a solution with a low entry barrier and low running costs.

This can be achieved by outsourcing the entire activity, i.e. by handing over the data to trusted digital repositories. While this is definitely a feasible option, some institutions as well as individuals are weary about handing over sensitive digital objects to third parties. Thus, as a complementary strategy, we would like to see systems emerge that remove the complexity of preservation activities while ensuring that the objects themselves may remain with the owner [RBK+10]. To address these needs, we need largely hassle-free, automated solutions that do not require manual intervention. Small, automated solutions such as the home archiving tool HOPPPLA [SMSR08, GSPR10] can partly address this need. However, to be trustworthy, these systems need to be complemented by accountable planning decisions. By offering Planning as a Service, it would be possible to outsource only the complexity induced by the expertise required to address the challenges of digital preservation, and deliver this expertise to the customers in a packaged, executable form in the shape of executable preservation plans.

7.3.7 Scale up: Automated scalable preservation planning

Where are we now? Using the work presented in this thesis, we can create solid, well-founded, well-documented and trustworthy preservation plans that treat a certain part of a large repository. These plans are able to treat a well-defined set of objects. They need to be monitored manually, they are not easily applicable to heterogeneous holdings, and constructing them still involves considerable effort. On the other hand, monitoring the user communities and changing technology is generally, if at all, done by publishing quarterly white papers that discuss the findings of a person analysing some trends, trying to draw conclusions and give recommendations on which file formats might be more appropriate than others. These are based on rather vague and implicit assumptions, read by a certain audience and can hardly be used as a basis for solid decision making, since the information they provide is not specific and well-founded enough to derive concrete actions, and they are not available in any machine-readable form.

The challenge facing institutions today is to make digital preservation scale up to their expected volumes of Petabytes of data. Current efforts directed towards leveraging grid technologies promise a step forward into that direction. But fundamentally, for a system to be truly operational on a large scale, all components involved need to scale up. Scalability for handling massive amounts of data can be achieved by state of the art grid technologies. However, even if a grid would be able to migrate millions of objects within seconds and generate QA data on-the-fly with minimal overhead, we have no way of planning, monitoring, and operating a repository on a Terabyte-scale as of today. Only scalable monitoring and decision making enables automated, large-scale operation of scalable tools and systems by scaling up the decision making and QA structures, policies, processes, and procedures for monitoring and action.

There is a bottleneck of processing information required for decision making and automating the now-manual steps such as monitoring, measurements, information reuse and the knowledge base by integrating existing and evolving information sources and measurements. Planning processes and plans need to become automatically traceable and auditable, applicable to heterogeneous content, scalable, and cost-efficient.

- *Traceability and auditability*: In order to ensure authenticity and integrity in the light of compliance requirements and trust, increased automation must be accompanied by audit trails of evidence leading to actions. This evidence must be connected
  - to policies and decision constraints posed by the organisation,
  - to the content held by an organisation (the collection profiles), and

  - to systems operation (through QoS monitoring and SLAs).

- *Applicability*: Preservation planning must be able to treat complex and compound objects as well as heterogeneous content.

- *Scalability and cost-efficiency*: We must increase the automation in decision making up to a point where planning is not a step-by-step construction of actions, but a definition of rules, preferences and constraints that lead to an appropriate recommendation and subsequent deployment of automated actions.

Completing the cycle, this also means that policies and plans need not only to be monitored, but also evolve along the lifecycle of digital content, according to a dynamically changing environment. Plan enactment and continuous operation needs to be monitored continuously on all levels; measurements need to be collected and analysed automatically to trigger appropriate events; and changes in the environment must be detected and lead to automated notifications that trigger decision making.

Ultimately, planning and watch needs to emerge from one-off decision making procedures to a continuous, and continuously optimising, management activity.

Bibliography

[AC01] Carina Alves and Jaelson Castro. CRE: A systematic method for COTS components selection. In *XV Brazilian Symposium on Software Engineering (SBES)*, Rio de Janeiro, Brazil, 2001.

[ADM+08] P. Ayris, R. Davies, R. McLeod, R. Miao, H. Shenton, and P. Wheatley. The LIFE2 final project report. *LIFE Project*, 2008. `http://eprints.ucl.ac.uk/11758/`.

[Ado10] Adobe. Macromedia Director and Adobe Shockwave Player: FAQ, accessed May 2010. `http://www.adobe.com/products/director/special/crossproduct/faq.html`.

[AF10] Caroline R. Arms and Carl Fleischhauer. *The Digital Formats Web site*. The Library of Congress, accessed May 2010. `http://www.digitalpreservation.gov/formats/`.

[AGCP09] Stavros Angelis, Dimitris Gavrilis, Panos Constantopoulos, and Christos Papatheodorou. A digital library service for the small. In *DigCCurr 2009: Digital Curation Practice, Promise and Prospects*, 2009.

[AHJ+08] Brian Aitken, Petra Helwig, Andrew Jackson, Andrew Lindley, Eleonora Nicchiarelli, and Seamus Ross. The Planets Testbed: Science for digital preservation. *code4lib Journal*, 3, June 2008. `http://journal.code4lib.org/articles/83`.

[Alv03] Carina Alves. *Component-Based Software Quality*, volume 2693 of *LNCS*, chapter COTS-Based Requirements Engineering, pages 21–39. Springer Berlin Heidelberg, 2003.

[ANS06] ANSI/NISO. *ANSI/NISO Z39.87 - Data Dictionary - Technical Metadata for Digital Still Images*. NISO Standard, December 2006.

[bam07] Archiving the Avant Garde: Documenting and Preserving Digital / Variable Media Art, accessed May 2007. `http://www.bampfa.berkeley.edu/about/avantgarde`.

[Bau88] Benjamin Bauermeister. *A Manual of Comparative Typography*. Van Nostrand Reinhold, 1988.

[BCH+07] Tim Brody, Leslie Carr, Jessie M.N. Hey, Adrian Brown, and Steve Hitchcock. PRONOM-ROAR: Adding format profiles to a repository registry to inform preservation services. *International Journal of Digital Curation*, 2(2):3–19, November 2007.

[Bea99] David Bearman. Reality and chimeras in the preservation of electronic records. *D-Lib Magazine*, 5(4), April 1999.

[Bec08] Christoph Becker. Automating the preservation planning process: An extensible evaluation framework for digital preservation. In Andreas Rauber and Jan Paralic, editors, *Workshop on Data Analysis (WDA 2008)*, pages 27–42, Dedinky, Slovakia, June 2008.

[Bes01] Howard Besser. Longevity of electronic art. *submitted to International Cultural Heritage Informatics Meeting*, February 2001.

[BFK+08] Christoph Becker, Miguel Ferreira, Michael Kraxner, Andreas Rauber, Ana Alice Baptista, and José Carlos Ramalho. Distributed preservation services: Integrating planning and actions. In Birte Christensen-Dalsgaard, Donatella Castelli, Bolette Ammitzbøll Jurik, and Joan Lippincott, editors, *Research and Advanced Technology for Digital Libraries. Proceedings of the 12th European Conference on Digital Libraries (ECDL 2008)*, volume LNCS 5173 of *Lecture Notes in Computer Science*, pages 25–36, Aarhus, Denmark, September 14–19 2008. Springer Berlin/Heidelberg.

[BHSS06] Paul Baker, Mark Harman, Kathleen Steinhöfel, and Alexandros Skaliotis. Search based approaches to component selection and prioritization for the next release problem. In *Proceedings of the 22nd IEEE International Conference on Software Maintenance (ICSM 2006)*, pages 176–185. IEEE Computer Society, 2006.

[BKG+09] Christoph Becker, Hannes Kulovits, Mark Guttenbrunner, Stephan Strodl, Andreas Rauber, and Hans Hofman. Systematic planning for digital preservation: Evaluating potential strategies and building preservation plans. *International Journal on Digital Libraries (IJDL)*, December 2009. `http://dx.doi.org/10.1007/s00799-009-0057-1`.

[BKK+09a] Christoph Becker, Hannes Kulovits, Michael Kraxner, Riccardo Gottardi, and Andreas Rauber. An extensible monitoring framework for measuring and evaluating tool performance in a service-oriented architecture. In Oscar Díaz Martin Gaedke, Michael Grossniklaus, editor, *Proceedings of the 9th International Conference on Web Engineering (ICWE 2009)*, volume LNCS 5648, pages 221–235. Springer, 2009.

[BKK+09b] Christoph Becker, Hannes Kulovits, Michael Kraxner, Riccardo Gottardi, Andreas Rauber, and Randolph Welte. Adding quality-awareness to evaluate migration web-services and remote emulation for digital preservation. In Maristella Agosti, Jose Borbinha, Sarantos Kapidakis, Christos Papatheodorou, and Giannis Tsakonas, editors, *Research and Advanced Technology for Digital Libraries. Proceedings of the 13th European Conference on Digital Libraries (ECDL 2009)*, volume 5714 of *LNCS*, pages 39–50. Springer, September 2009.

[BKKR07] Christoph Becker, Guenther Kolar, Josef Kueng, and Andreas Rauber. Preserving interactive multimedia art: A case study in preservation planning. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers. Proceedings of the Tenth Conference on Asian Digital Libraries (ICADL 2007)*, volume 4822/2007 of *Lecture Notes in Computer Science*, pages 257–266, Hanoi, Vietnam, December 10-13 2007. Springer Berlin / Heidelberg.

[BKL+09] Matt Blaze, Sampath Kannan, Insup Lee, Oleg Sokolsky, Jonathan M. Smith, Angelos D. Keromytis, and Wenke Lee. Dynamic trust management. *IEEE Computer*, 42(2):44–52, February 2009.

[BKR10] Christoph Becker, Hannes Kulovits, and Andreas Rauber. Trustworthy preservation planning with plato. *ERCIM News*, 80:24–25, January 2010.

[BKRH08] Christoph Becker, Hannes Kulovits, Andreas Rauber, and Hans Hofman. Plato: A service oriented decision support system for preservation planning. In *Proceedings of the 8th ACM IEEE Joint Conference on Digital Libraries (JCDL 2008)*, 2008.

[Bla90] Sandra Blakeslee. Lost on earth: Wealth of data found in space. *New York Times*, March 20 1990. `http://www.nytimes.com/1990/03/20/science/lost-on-earth-wealth-of-data-found-in-space.html?sec=&spon=&pagewanted=all`.

[Blo33] Leonard Bloomfield. *Language*. Allen and Unwin, New York, 1933.

[BLPW99] Kent Blackburn, Albert Lazzarini, Tom Prince, and Roy Williams. *High-Performance Computing and Networking*, chapter XSIL: Extensible scientific interchange language, pages 513–524. Springer Berlin / Heidelberg, 1999.

[BR07] Christoph Becker and Andreas Rauber. *Langfristige Archivierung digitaler Fotografien*. December 2007. `http://www.ifs.tuwien.ac.at/dp/fotostudie/`.

[BR09] Christoph Becker and Andreas Rauber. Requirements modelling and evaluation for digital preservation: A COTS selection method based on controlled experimentation. In *Proceedings of the 24th ACM Symposium on Applied Computing (SAC 2009)*, Honolulu, Hawaii, USA, 2009. ACM Press.

[BR10] Christoph Becker and Andreas Rauber. Improving component selection and monitoring with controlled experimentation and automated measurements. *Information and Software Technology*, 52(6):641–655, June 2010.

[BRH+08a] Christoph Becker, Andreas Rauber, Volker Heydegger, Jan Schnasse, and Manfred Thaller. A generic XML language for characterising objects to support digital preservation. In *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC 2008)*, volume 1, pages 402–406, Fortaleza, Brazil, March 16-20 2008. ACM.

[BRH+08b] Christoph Becker, Andreas Rauber, Volker Heydegger, Jan Schnasse, and Manfred Thaller. Systematic characterisation of objects in digital preservation: The extensible characterisation languages. *Jour-*

nal of Universal Computer Science, 14(18):2936–2952, 2008. `http://www.jucs.org/jucs_14_18/systematic_characterisation_of_objects`.

[BSN+07] Christoph Becker, Stephan Strodl, Robert Neumayer, Andreas Rauber, Eleonora Nicchiarelli Bettelli, and Max Kaiser. Long-term preservation of electronic theses and dissertations: A case study in preservation planning. In *Proceedings of the 9th Russian Conference on Digital Libraries (RCDL 2007)*, Pereslavl, Russia, October 2007. `http://rcdl2007.pereslavl.ru/en/program.shtml`.

[BW04] M. Beckerle and M. Westhead. GGF DFDL Primer. Technical report, Global Grid Forum Data Format Description Language Working Group, 2004.

[CCMP08] Vittorio Cortellessa, Ivica Crnkovic, Fabrizio Marinelli, and Pasqualina Potena. Experimenting the automated selection of COTS components based on cost and system requirements. *Journal of Universal Computer Science*, 14(8):1228–1255, 2008. `http://www.jucs.org/jucs_14_8/experimenting_the_automated_selection`.

[Cen07] Center for Research Libraries. Ten principles, January 2007. `http://www.crl.edu/archiving-preservation/digital-archives/metrics-assessing-and-certifying/core-re`, accessed March 8, 2010.

[CFQ07] Juan Pablo Carvallo, Xavier Franch, and Carme Quer. Determining criteria for selecting software components: Lessons learned. *IEEE Software*, 24(3):84–94, May-June 2007.

[CFQ08] Juan P. Carvallo, Xavier Franch, and Carme Quer. Requirements engineering for COTS-based software systems. In *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC 2008)*, pages 638–644, New York, NY, USA, 2008. ACM.

[Con08] Consultative Committee for Space Data Systems. Metrics for digital repository audit and certification. Draft White Book, May 2008. `http://wiki.digitalrepositoryauditandcertification.org/`.

[CPIZ07] O. Chum, J. Philbin, M. Isard, and A. Zisserman. Scalable near identical image and shot detection. In *Proceedings of*

the International Conference on Image and Video Retrieval, Amsterdam, The Netherlands, 2007.

[CPV03] Alejandra Cechich, Mario Piattini, and Antonio Vallecillo, editors. *Component-Based Software Quality*. Springer, 2003.

[CX05] E. Rodney Canfield and Guangming Xing. Approximate XML document matching. In *Proceedings of the 20th ACM symposium on Applied computing (SAC 2005)*, pages 787–788, New York, NY, USA, 2005. ACM Press.

[CYWM03] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. Vips: a vision-based page segmentation algorithm. Technical report, Microsoft Research, 2003.

[Deb60] G. Debreu. Review of R.D. Luce Individual choice behavior. *American Economic Review*, 50:186–188, 1960.

[Dep01] Alain Depocas. Digital preservation: Recording the Recoding. The Documentary Strategy. In *Ars Electronica 2001: Takeover. Who's doing the Art of Tomorrow?*, 2001. `http://www.aec.at/festival2001/texte/depocas_e.html`.

[DF09] Angela Dappert and Adam Farquhar. Significance is in the eye of the. stakeholder. In Maristella Agosti, Jose Borbinha, Sarantos Kapidakis, Christos Papatheodorou, and Giannis Tsakonas, editors, *Research and Advanced Technology for Digital Libraries. Proceedings of the 13th European Conference on Digital Libraries (ECDL 2009)*, volume 5714 of *LNCS*, pages 39–50. Springer, September 2009.

[Dig02] Digital Preservation Testbed Project. XML and digital preservation. Technical report, 2002. `http://www.digitaleduurzaamheid.nl/bibliotheek/docs/white-paper_xml-en.pdf`.

[DIJ04] Alain Depocas, Jon Ippolito, and Caitlin Jones, editors. *Permanence Through Change: The Variable Media Approach*. Guggenheim Museum Publications, February 2004.

[Dod02] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[Doy05] John R. Doyle. Evaluating the IBM and HP/PANOSE font classification systems. *Online Information Review*, 29(5):468–482, 2005.

[DS05] Schahram Dustdar and Wolfgang Schreiner. A survey on web services composition. *International Journal of Web and Grid Services*, 1:1–30, 2005.

[DSS07] Susanne Dobratz, Astrid Schoger, and Stefan Strathmann. The nestor catalogue of criteria for trusted digital repository evaluation and certification. *Journal of Digital Information*, 8(2), 2007. `http://journals.tdl.org/jodi/article/viewArticle/199/180`.

[DWB02] Luis Martín Díaz, Erik Wüstner, and Peter Buxmann. Inter-organizational document exchange: Facing the conversion problem with XML. In *Proceedings of the 2002 ACM symposium on Applied computing (SAC 2002)*, pages 1043–1047, New York, NY, USA, 2002. ACM Press.

[EDG+02] M. K. Evi, M. Diligenti, M. Gori, M. Maggini, and V. Milutinovi. Recognition of common areas in a web page using visual information: a possible application in a page classification. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, 2002.

[EMT07] Abdelkarim Erradi, Piyush Maheshwari, and Vladimir Tosic. Ws-policy based monitoring of composite web services. In *Proceedings of the Fifth European Conference on Web Services (ECOWS 2007)*, pages 99–108, Washington, DC, USA, 2007. IEEE Computer Society.

[Erp03] Erpanet. *Digital Preservation Policy Tool*, September 2003. `http://www.erpanet.org/guidance/docs/ERPANETPolicyTool.pdf`.

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

[FBR06] Miguel Ferreira, Ana Alice Baptista, and Jose Carlos Ramalho. A foundation for automatic digital preservation. *Ariadne*, 48, July 2006.

[FBR07] Miguel Ferreira, Ana Alice Baptista, and Jose Carlos Ramalho. An intelligent decision support system for digital preservation. *International Journal on Digital Libraries*, 6(4):295–304, July 2007.

[FC03] X. Franch and J.P. Carvallo. Using quality models in software package selection. *IEEE Software*, 20(1):34–41, Jan/Feb 2003.

[FG05] K. Fisher and R. Gruber. A domain-specific language for processing ad hoc data. In *Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation*, pages 295–304, 2005.

[FHY07] Adam Farquhar and Helen Hockx-Yu. Planets: Integrated services for digital preservation. *International Journal of Digital Curation*, 2(2):88–99, November 2007.

[Flo08] Florida Centre for Library Automation. Recommended data formats for preservation purposes in the FCLA digital archive, August 2008. `http://www.fcla.edu/digitalArchive/pdfs/recFormats.pdf`.

[FMW06] K. Fisher, Y. Mandelbaum, and D. Walker. The next 700 data description languages. In *Conference record of the 33rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 2–15, 2006.

[FTRK09] Adam Field, David Tarrant, Andreas Rauber, and Hannes Kulovits. Digital preservation: Logical and bit-stream preservation using plato, eprints and the cloud. In *13th European Conference on Digital Libraries (ECDL)*, Corfu, Greece, September 2009. `http://eprints.ecs.soton.ac.uk/17962/`.

[GAO95] David Garlan, Robert Allen, and John Ockerbloom. Architectural mismatch or why it's hard to build systems out of existing parts. *International Conference on Software Engineering*, page 179, 1995.

[GAO09] D. Garlan, R. Allen, and J. Ockerbloom. Architectural mismatch: Why reuse is still so hard. *Software, IEEE*, 26(4):66–69, July-Aug. 2009.

[Gau07] Sharon Gaudin. The digital universe created 161 exabytes of data last year. *InformationWeek*, March 7 2007. `http://www.informationweek.com/news/internet/search/showArticle.jhtml?articleID=197800880`, accessed January 30, 2010.

[GBC+06] Félix García, Manuel F. Bertoa, Coral Calero, Antonio Vallecillo, Francisco Ruíz, Mario Piattini, and Marcela Gen-

ero. Towards a consistent terminology for software measurement. *Information and Software Technology*, 48(8):631 – 644, 2006.

[GBR08] Mark Guttenbrunner, Christoph Becker, and Andreas Rauber. Evaluating strategies for the preservation of console video games. In *Proceedings of the Fifth international Conference on Preservation of Digital Objects (iPRES 2008)*, London, UK, September 2008.

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

[GCFQ04] Gemma Grau, Juan Pablo Carvallo, Xavier Franch, and Carme Quer. DesCOTS: A software system for selecting COTS components. In *Proceedings of the 30th EUROMICRO Conference (EUROMICRO 2004)*, pages 118–126, Washington, DC, USA, 2004. IEEE Computer Society.

[GCMC02] X.-D. Gu, J. Chen, W.-Y. Ma, and G.-L. Chen. Visual based content understanding towards web adaptation. In *Second International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH2002)*, 2002.

[Ger90] John S. Gero. Design prototypes: A knowledge representation schema for design. *AI Magazine*, 11(4), Winter 1990. `http://www.aaai.org/ojs/index.php/aimagazine/article/view/854`.

[GKM09] Stephen Grace, Gareth Knight, and Lynne Montague. *InSPECT Final Report*. InSPECT (Investigating the Significant Properties of Electronic Content over Time), December 2009. `http://www.significantproperties.org.uk/inspect-finalreport.pdf`.

[GR10] John Gantz and David Reinsel. *The Digital Universe Decade – Are You Ready?* IDC, May 2010. `http://idcdocserv.com/925`.

[Gra00] Stewart Granger. Emulation as a digital preservation strategy. *D-Lib Magazine*, 6(10), October 2000.

[GSE00] A.J. Gilliland-Swetland and P.B. Eppard. Preserving the authenticity of contingent digital objects: The InterPARES project. *D-Lib Magazine*, 6(7/8), July-August 2000. `http://www.dlib.org/dlib/july00/eppard/07eppard.html`.

[GSPR10] Michael Greifeneder, Stephan Strodl, Petar Petrov, and Andreas Rauber. HOPPLA - archiving system for small institutions. *ERCIM News*, 80, 2010.

[HC03] Jane Hunter and Sharmin Choudhury. Implementing preservation strategies for complex multimedia objects. In *The Seventh European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2003)*, pages 473–486, Trondheim, Norway, August 17-22 2003.

[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*, pages 269–278, Tucson, Arizona, June 2004. ACM Press, New York.

[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.

[HJ01] Mark Harman and Bryan F. Jones. Search-based software engineering. *Information and Software Technology*, 43(14):833 – 839, 2001.

[Hor05] M. Horvath. Empfehlungen zum erzeugen archivierbarer dateien im format pdf. Technical report, Austrian National Library, August 2005. `http://www.onb.ac.at/files/ONB_PDF-Einstellungen_Distiller7_1-0.pdf` (in German).

[HPB+08] Hans Hofman, Planets-PP subproject, Christoph Becker, Stephan Strodl, Hannes Kulovits, and Andreas Rauber. Preservation plan template. Technical report, The Planets project, 2008. `http://www.ifs.tuwien.ac.at/dp/plato/docs/plan-template.pdf`.

[Hsi09] Nien-he Hsieh. Incommensurable values. Stanford Encyclopedia of Philosophy, May 2009. `http://plato.stanford.edu/entries/value-incommensurable/`.

[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.

[HYK08] Helen Hockx-Yu and Gareth Knight. What to preserve?: Significant properties of digital objects. *International Journal of Digital Curation*, 3(1), 2008.

[ISO99] ISO. *Information technology – Software product evaluation – Part 1: General overview (ISO/IEC 14598-1:1999).* International Standards Organization, 1999.

[ISO01] ISO. *Software Engineering – Product Quality – Part 1: Quality Model (ISO/IEC 9126-1).* International Standards Organization, 2001.

[ISO02] ISO. *Information technology – Multimedia content description interface – Part 1: Systems ISO/IEC 15938-1:2002.* International Standards Organization, 2002.

[ISO03] ISO. *Open archival information system – Reference model (ISO 14721:2003).* International Standards Organization, 2003.

[ISO04a] ISO. *Document management - Electronic document file format for long-term preservation - Part 1: Use of PDF 1.4 (PDF/A) (ISO/CD 19005-1).* International Standards Organization, 2004.

[ISO04b] ISO. *Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification (ISO/IEC 15948:2004).* International Standards Organization, 2004.

[ISO04c] ISO. *Information technology – JPEG 2000 image coding system – Part 1: Core coding system (ISO/IEC 15444-1:2004).* International Standards Organisation, 2004.

[ISO06] ISO. *Information technology – Open Document Format for Office Applications (ISO/IEC 26300:2006).* International Standards Organization, 2006.

[ISO07a] ISO. *Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Measurement reference model and guide (ISO/IEC 25020:2007).* International Standards Organisation, 2007.

[ISO07b] ISO. *Systems and software engineering – Measurement process (ISO/IEC 15939:2007).* International Standards Organisation, 2007.

[JB08] Maggie Jones and Neil Beagrie. *Preservation Management of Digital Materials: A Handbook*. Digital Preservation Coalition, London, UK, November 2008. `http://www.dpconline.org/vendor-reports/download-document/299-digital-preservation-handbook.html`, accessed May 2010.

[JMRIR07] Antonio Jiménez, Alfonso Mateos, Sixto Ríos-Insua, and Luis Carlos Rodríguez. Contracting cleaning services in a european public underground transportation company with the aid of a dss. *Decision Support Systems*, 43(4):1485 – 1498, 2007. Special Issue Clusters.

[Joh09] John Hutchins. Compendium of translation software, 15th edition. The European Association for Machine Translation, January 2009. `http://www.hutchinsweb.me.uk/Compendium-15.pdf`.

[Jon04] Caitlin Jones. Seeing double: Emulation in theory and practice. the Erl King case study. In *Electronic Media Group, Annual Meeting of the American Institute for Conservation of Historic and Artistic Works*. Variable Media Network, Solomon R. Guggenheim Museum, 2004.

[JS09] Anil S. Jadhav and Rajendra M. Sonar. Evaluating and selecting software packages: A review. *Information and Software Technology*, 51(3):555–563, 2009.

[KHL01] B.A. Kitchenham, R.T. Hughes, and S.G. Linkman. Modeling software measurement data. *IEEE Transactions on Software Engineering*, 27(9):788–804, Sep 2001.

[KL03] Alexander Keller and Heiko Ludwig. The WSLA framework: Specifying and monitoring service level agreements for web services. *Journal of Network and Systems Management*, 11(1):57–81, March 2003.

[Kon95] Jyrki Kontio. OTSO: a systematic process for reusable software component selection. Technical report, College Park, MD, USA, 1995.

[Kon96] Jyrki Kontio. A case study in applying a systematic method for COTS selection. In *Proceedings of the 18th International Conference on Software Engineering (ICSE-18)*, pages 201–209, 1996.

[KP09] Gareth Knight and Maureen Pennock. Data without meaning: Establishing the significant properties of digital research. *International Journal of Digital Curation*, 4(1):159–174, 2009.

[KR93] Ralph L. Keeney and Howard Raiffa. *Decisions with multiple objectives: preferences and value tradeoffs*. Cambridge University Press, 1993.

[Kra09] Michael Kraxner. Maintaining a preservation plan: Serialisation to XML as part of a documentation strategy. Bachelor thesis, May 2009. Vienna University of Technology.

[KRB+09] Hannes Kulovits, Andreas Rauber, Markus Brantl, Astrid Schoger, Tobias Beinert, and Anna Kugler. From TIFF to JPEG2000? Preservation planning at the Bavarian State Library using a collection of digitized 16th century printings. *D-Lib Magazine*, 15(11/12), November/December 2009. `http://dlib.org/dlib/november09/kulovits/11kulovits.html`.

[KSJ+09] Ross King, Rainer Schmidt, Andrew N. Jackson, Carl Wilson, and Fabian Steeg. The Planets Interoperability Framework: an infrastructure for digital preservation actions. In *Proceedings of the 13th European Conference on Digital Libraries (ECDL'2009)*, 2009.

[LB94] James R. Larus and Thomas Ball. Rewriting executable files to measure program behavior. *Software: Practice and Experience*, 24(2):197–218, 1994.

[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, Council on Library and Information Resources, June 2000.

[LRL05] Lucian Vlad Lita, Monica Rogati, and Alon Lavie. BLANC: learning evaluation metrics for MT. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HTL 2005)*, pages 740–747, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[Man10] Pat Manson. *Digital Preservation Research: An Evolving Landscape*, volume ERCIM News 80 of *ERCIM News*.

European Research Consortium for Informatics and Mathematics, 2010. `http://ercim-news.ercim.eu/images/stories/EN80/EN80-web.pdf`.

[McL96] I. McLean. Independence of irrelevant alternatives before arrow. *Mathematical Social Sciences*, 31(1), February 1996.

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

[Men02] Daniel A. Menascé. QoS issues in web services. *IEEE Internet Computing*, 6(6):72–75, 2002.

[MG09] João Miranda and Daniel Gomes. Trends in Web characteristics. In *7th Latin American Web Congress (LA-Web 2009)*, Merida, Mexico, November 2009.

[MH67] M. L. Mannheim and F. Hall. Abstract representation of goals: A method for making decisions in complex problems. In *Transportation: A service, proceedings of the sesquicentennial forum.* New York Academy of Sciences - American Society of Mechanical Engineers, 1967.

[MK10] Andrew McHugh and Leonidas Konstantelos. The art of preserving digital creativity in planets. *ERCIM News*, 80, January 2010.

[MKB09] Andrew McHugh, Leonidas Konstantelos, and Matthew Barr. Report on emerging digital art characterisation techniques. Technical Report PC5-D5, Planets Project Deliverable, November 2009. `http://www.planets-project.eu/docs/reports/Digital_Art_Characterisation_Techniques.pdf`.

[MN98] N.A. Maiden and C. Ncube. Acquiring COTS software selection requirements. *IEEE Software*, 15(2):46–56, Mar/Apr 1998.

[MRE07] Abdallah Mohamed, Guenther Ruhe, and Armin Eberlein. COTS selection: Past, present, and future. In *Proceedings of the 14th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2007)*, pages 103–114, 2007.

[MS04] E. Michael Maximilien and Munindar P. Singh. Toward autonomic web services trust and selection. In *Proceedings of*

the 2nd international conference on Service oriented computing (ICSOC 2004), pages 212–221, New York, NY, USA, 2004. ACM.

[MWS02] P. Mellor, P. Wheatley, and D.M. Sergeant. Migration on request, a practical technique for preservation. In M. Agosti and M.C. Thanos, editors, *Proceedings of the 6th European Conference on Digital Libraries (ECDL 2002)*, pages 516–526. Springer, 2002.

[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.

[ND02] Cornelius Ncube and John C. Dean. *COTS-Based Software Systems*, volume 2255 of *LNCS*, chapter The Limitations of Current Decision-Making Techniques in the Procurement of COTS Software Components, pages 176–187. Springer Berlin / Heidelberg, 2002. `http://www.springerlink.com/content/t8pp09vhjdncg913/`.

[Nei08] Neil Beagrie, Najla Semple, Peter Williams, and Richard Wright. Digital Preservation Policies Study. Technical report, Charles Beagrie Limited, October 2008.

[nes06] nestor Working Group -Trusted Repositories Certification. Catalogue of Criteria for Trusted Digital Repositories. Technical report, nestor - Network of Expertise in long-term STORage, Frankfurt am Main, June 2006. Version 1.

[NM99] Cornelius Ncube and Neil. A. M. Maiden. PORE: Procurement-oriented requirements engineering method for the component-based systems engineering development paradigm. In *Development Paradigm. International Workshop on Component-Based Software Engineering*, pages 1–12, 1999.

[NS07a] Nicholas Nethercote and Julian Seward. Valgrind: a framework for heavyweight dynamic binary instrumentation. *SIGPLAN Not.*, 42(6):89–100, 2007.

[NS07b] Thomas Neubauer and Christian Stummer. Interactive decision support for multiobjective cots selection. In *Proceedings of the 40th Annual Hawaii International Conference*

on System Sciences (HICSS 2007), page 283b, Washington, DC, USA, 2007. IEEE Computer Society.

[Ogd98] Sherelyn Ogden. *Preservation Planning: Guidelines for Writing a Long-Range Plan.* American Association of Museums, 1998.

[Ogd10] Sherelyn Ogden. *What Is Preservation Planning?* Northeast Document Conservation Center, accessed January 2010. `http://www.nedcc.org/resources/leaflets/1Planning_and_Prioritizing/01WhatIsPreservationPlanning.php`.

[OPCDNK00] Michael Ochs, Dietmar Pfahl, Gunther Chrobok-Diening, and Beate Nothhelfer-Kolb. A COTS acquisition process: Definition and application experience. In *Proc. of the ESCOM-SCOPE 2000*, pages 353–343, Munich, Germany, 2000. Shaker Publ.

[OPCDNK01] M. Ochs, D. Pfahl, G. Chrobok-Diening, and B. Nothhelfer-Kolb. A method for efficient measurement-based COTS assessment and selection method description and evaluation results. In *Proceedings of the Seventh International Software Metrics Symposium (METRICS 2001)*, pages 285–296, 2001.

[Pow10] Alan Powell. *Data Format Description Language (DFDL) v1.0 Core Specification (Internal Committee Working Document) version 039.* Open Grid Forum Data Format Description Language Working Group, February 2010. `http://forge.gridforum.org/sf/go/doc15889?nav=1`.

[PRD07] C. Platzer, F. Rosenberg, and S. Dustdar. *Securing Web Services: Practical Usage of Standards and Specifications*, chapter Enhancing Web Service Discovery and Monitoring with Quality of Service Information. Idea Publishing Inc, 2007.

[PRS09] Anna Perini, Filippo Ricca, and Angelo Susi. Tool-supported requirements prioritization: Comparing the AHP and CBRank methods. *Information and Software Technology*, 51(6):1021–1032, 2009.

[PRWZ02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002)*,

pages 311–318, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[QFLP05] Carme Quer, Xavier Franch, and Xavier Lopez-Pelegrin. DesCOTS-EV: A tool for the evaluation of COTS components. In *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE 2005)*, pages 457–460, Washington, DC, USA, 2005. IEEE Computer Society.

[QFLP06] Carme Quer, Xavier Franch, and Xavier Lopez-Pelegrin. DesCOTS-SL: A tool for the selection of COTS components. In *Proceedings of the 14th IEEE International Requirements Engineering Conference (RE 2006)*, pages 358–359, Washington, DC, USA, 2006. IEEE Computer Society.

[Ran03] Shuping Ran. A model for web services discovery with QoS. *SIGecom Exchanges*, 4(1):1–10, 2003.

[Rau04] Carl Rauch. Preserving digital entities: A framework for choosing and testing preservationstrategies. Master's thesis, Vienna University of Technology, 2004.

[Ray73] Paramesh Ray. Independence of irrelevant alternatives. *Econometrica*, 41(5), September 1973.

[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, The Hague, Netherlands, 1999.

[RBK+10] Andreas Rauber, Christoph Becker, Hannes Kulovits, Michael Greifeneder, Petar Petrov, and Stephan Strodl. Digital preservation: From large-scale institutions via smes to individual users. In *International Conference on Digital Libraries (ICDL 2010)*, New Delhi, India, February 2010.

[RBRH+08] Colin Rosenthal, Asger Blekinge-Rasmussen, Jan Hutar, Andrew McHugh, Stephan Strodl, Emily Witham, and Seamus Ross. *Repository Planning Checklist and Guidance*. HATII at the University of Glasgow, 2008.

[RFC94] Richard Roth, Frank Field, and Joel Clark. Materials selection and multi-attribute utility analysis. *Journal of Computer-Aided Materials Design*, 1:325–342, 1994.

[RLG02] RLG/OCLC Working Group on Digital Archive Attributes. *Trusted Digital Repositories: Attributes and Responsibilities*. Research Libraries Group, 2002.

[RM06] Seamus Ross and Andrew McHugh. The role of evidence in establishing trust in repositories. *D-Lib Magazine*, 12(7/8), 2006.

[Rol99] C. Rolland. Requirements engineering for COTS based systems. *Information and Software Technology*, 41:985–990, 1999.

[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.

[Saa90] Thomas L. Saaty. How to make a decision: the Analytic Hierarchy Process. *European Journal of Operational Research*, 48(1):9–26, 1990.

[SBD+09] Pauline Sinclair, Clive Billenness, James Duckworth, Adam Farquhar, Jane Humphreys, Lewis Jardine, Ann Keen, and Robert Sharpe. Are you ready? Assessing whether organisations are prepared for digital preservation. In *Proceedings of the Sixth international Conference on Preservation of Digital Objects (iPRES 2009)*, 2009.

[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 *Revised Selected Papers of the 1st International DELOS Conference*, pages 238–247, Pisa, Italy, February 13-14 2007. Springer.

[SBNR07] Stephan Strodl, Christoph Becker, Robert Neumayer, and Andreas Rauber. How to choose a digital preservation strategy: Evaluating a preservation planning procedure. In *Pro-*

ceedings of the 7th ACM IEEE Joint Conference on Digital Libraries (JCDL 2007), pages 29–38, June 2007.

[SBR09] Stephan Strodl, Christoph Becker, and Andreas Rauber. *Handbook of Research on Digital Libraries: Design, Development, and Impact*, chapter Digital preservation, pages 431–440. Information Science Reference, 2009. ISBN: 978-159904879-6.

[SGP09] MB Saad, S Gançarski, and Z Pehlivan. A novel web archiving approach based on visual pages analysis. In *9th International Web Archiving Workshop (IWAW 2009)*, Corfu, Greece, 2009.

[SKS+09] Rainer Schmidt, Ross King, Fabian Steeg, Peter Melms, Andrew Jackson, and Carl Wilson. A framework for distributed preservation workflows. In *Proceedings of the Sixth international Conference on Preservation of Digital Objects (iPRES 2009)*, 2009.

[SMNBC09] H. Skogsrud, H.R. Motahari-Nezhad, B. Benatallah, and F. Casati. Modeling trust negotiation for web services. *Computer*, 42(2):54–61, Feb. 2009.

[SMSR08] Stephan Strodl, Florian Motlik, Kevin Stadler, and Andreas Rauber. Personal & SOHO archiving. In *Proceedings of the Joint Conference on Digital Libraries (JCDL 2008)*, 2008.

[SR08] Stephan Strodl and Andreas Rauber. Preservation planning in the OAIS model. *New Technology of Library and Information Service*, (1):61–68, Januar 2008.

[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.

[Sta04] Andreas Stanescu. Assessing the durability of formats in a digital preservation environment: The INFORM methodology. *D-Lib Magazine*, 10(11), November 2004.

[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.

[Ter09] Sotirios Terzis. The many faces of trust. *IEEE Computing Now*, April 2009. `http://www2.computer.org/portal/web/computingnow/archive/april2009`.

[Tes01] Digital Preservation Testbed. Migration: Context and current status. White paper, National Archives and Ministry of the Interior and Kingdom Relations, 2001.

[TGRS04] M. Tian, A. Gramm, H. Ritter, and J. Schiller. Efficient selection and monitoring of QoS-aware web services with the WS-QoS framework. *Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on*, pages 152–158, Sept. 2004.

[THC09] David Tarrant, Steve Hitchcock, and Les Carr. Where the Semantic Web and Web 2.0 meet format risk management: P2 registry. In *The Sixth International Conference on Preservation of Digital Objects (iPres 2009)*, 2009.

[The07] The 100 Year Archive Task Force. The 100 year archive requirements survey. `http://www.snia.org/forums/dmf/programs/ltacsi/100_year/`, 2007.

[The08] The Library of Congress. Preferences in summary for textual content. Website, last updated 2008. `http://www.digitalpreservation.gov/formats/content/text_preferences.shtml`.

[Thi02] Kenneth Thibodeau. Overview of technological approaches to digital preservation and challenges in coming years. In *The State of Digital Preservation: An International Perspective*, Washington, D.C., July 2002. Council on Library and Information Resources. `http://www.clir.org/pubs/reports/pub107/thibodeau.html`.

[TO07] The Center for Research Libraries (CRL) and Online Computer Library Center, Inc.(OCLC). Trustworthy Repositories Audit & Certification: Criteria and Checklist (TRAC). Technical report, CRL and OCLC, February 2007.

[Tod09] Malcolm Todd. Technology watch report: File formats for preservation. Technical report, The National Archives, October 2009.

[TSSM06] Tara D. Talbott, Karen L. Schuchardt, Eric G. Stephan, and James D. Myers. *Provenance and Annotation of Data*,

volume 4145, chapter Mapping Physical Formats to Logical Models to Extract Data and Metadata: The Defuddle Parsing Engine, pages 73–81. Springer Berlin / Heidelberg, 2006.

[Ull57] Stephen Ullmann. *Principles of Semantics*. Blackwell Oxford, 1957.

[Ull62] Stephen Ullmann. *Semantics: An introduction to the science of meaning*. Blackwell Oxford, 1962.

[UNE03] UNESCO. UNESCO charter on the preservation of digital heritage. Adopted at the 32nd session of the General Conference of UNESCO, October 17, 2003. `http://portal.unesco.org/ci/en/files/13367/10700115911Charter_en.pdf/Charter_en.pdf`.

[Var10] Variable Media Network. Variable media questionnaire, accessed May 2010. `http://www.variablemedia.net/e/welcome.html`.

[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.

[VL00] Peter Varian and Hal R. Lyman. Reprint: How much information? *The Journal of Electronic Publishing*, 6(6), December 2000. `http://dx.doi.org/10.3998/3336451.0006.204`.

[vL01] Axel van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE 2001)*, pages 249–263, Toronto, Canada, 2001.

[vNM44] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[VRA08] K. Vijayalakshmi, N. Ramaraj, and R. Amuthakkannan. Improvement of component selection process using genetic algorithm for component-based software development. *International Journal of Information Systems and Change Management*, 3(1):63–80, 2008.

[WB08] Kam Woods and Geoffrey Brown. Migration performance for legacy data access. *International Journal of Digital Curation*, 3(2), 2008.

[Web05] Colin Webb. *Guidelines for the Preservation of Digital Heritage*. Information Society Division United Nations Educational, Scientific and Cultural Organization (UNESCO) – National Library of Australia, 2005. `unesdoc.unesco.org/images/0013/001300/130071e.pdf`.

[WSA+01] P. Weirich, B. Skyrms, E.W. Adams, K. Binmore, J. Butterfield, P. Diaconis, and W.L. Harper. *Decision Space: Multidimensional Utility Analysis*. Cambridge University Press, 2001.

[WW05] N. Wickramage and S. Weerawarana. A benchmark for web service frameworks. *Services Computing, 2005 IEEE International Conference on*, 1:233–240 vol.1, July 2005.

[YBBP05] Y. Yang, Jesal Bhuta, B. Boehm, and D.N. Port. Value-based processes for COTS-based applications. *Software, IEEE*, 22(4):54–62, July-Aug. 2005.

[YCG08] Tim Au Yeung, Sheelagh Carpendale, and Saul Greenberg. Preservation of art in the digital realm. In *Proceedings of The Fifth International Conference on Preservation of Digital Objects (iPRES 2008)*, pages 32–29, 2008.