Diplomarbeit

Preserving Digital Entities

A Framework for Choosing and Testing Preservation Strategies

Ausgeführt am Institut für Software Technologie und Interaktive Systeme
der technischen Universität Wien

unter der Anleitung von
ao. Univ. Prof. Dr. Andreas Rauber

durch
Carl Rauch, Bakk.
Waldheimatweg 33
8043 Graz
carl.rauch@gmx.at

Datum: November 17, 2004 Unterschrift:

Contents

Zusammenfassung

Die langfristige Speicherung digitaler Objekte hat in den letzten Jahren immer mehr an Bedeutung gewonnen. Bibliotheken, Museen, die öffentliche Verwaltung und zunehmend auch Unternehmen suchen nach Lösungen, ihre digital vorhandenen Informationen mit allen relevanten Eigenschaften für die nächsten Generationen zu bewahren. Mit dieser Diplomarbeit sollen zwei Beiträge zur Forschung in diesem Bereich geleistet werden.

Der erste besteht im Aufbau einer Testumgebung, in der zahlreiche digitale Objekte in verschiedensten Formaten gespeichert werden. Damit kann die Auswirkung unterschiedlicher Speichermethoden auf die Eigenschaften der digitalen Objekte bestimmt werden. In dieser Arbeit wird ein Gerüst für eine derartige Testumgebung vorgeschlagen und implementiert.

Den zweiten Beitrag zur langfristigen Datenspeicherung stellt die Einführung eines Werkzeugs dar, mit dem verschiedene Speichermethoden miteinander verglichen werden können. Das geschieht auf Basis der *Nutzwertanalyse* mit Hilfe einer breiten Hierarchie von Kriterien, welche die Entscheidung beeinflussen und die viele verschiedene Aspekte der langfristigen Datenspeicherung abdecken, wie die Art der Daten, die Anforderungen der Nutzer oder die anfallenden Kosten. Die Verwendbarkeit des theoretischen Modells wird mit Hilfe zweier Beispiele praktisch dargestellt, im ersten wird der gesamte Prozess umgesetzt, im zweiten nur der wichtigste Teil - die Analyse der relevanten Kriterien.

Abstract

The long-term preservation of digital objects has become increasingly relevant. Libraries, public institutions and museums, but also companies are requesting solutions to store their digital files with all relevant contents and attributes for the future. This master thesis makes two contributions to the research in digital preservation.
The first attempt is the creation of a testbed which stores many files in different file formats. These files can be used to evaluate the impact of preservation solutions. In this paper an environment for storing and describing files is being suggested and implemented.
The second contribution is made by presenting a framework which is based on *Utility Analysis* for evaluating different preservation solutions. The application of a detailed hierarchy of objectives, considering the individual requirements of the user, will allow a reasonable and clear decision for a specific preservation solution, which can be supported with arguments. The theoretic framework is evaluated in two casestudies. For the first one the whole process is being realized, for the second example only the major part of the analysis, the objective tree, is treated in detail.

1 Introduction

The present-day rapid expansion of digital data creation and the trend towards digitally saving files and documents leads to an increasing demand for robust and trustworthy digital archives. Research in the preservation field has been focussing on storage media. CD-Rs with a reported lifetime of more than a hundred years [46], or systems that automatically migrate data to the most adequate storage media are available [14].

In the last couple of years a second issue became urgent - the preservation of digital files in different file formats. Since file formats are changing rapidly, it is nowadays doubtful whether it will be possible to reopen a file 10 years after its creation without loosing parts of its characteristics. Take for typical examples changes in the appearance, or the inability to interpret certain character encodings, makros and other. In some cases reopening a file might lead to a long line of uninterpretable characters or a nonsense bit stream. Some scientists already predict that our period will in the future be seen as an 'Age of Oblivion' [38] instead of an information age, due to the inability to read and interpret the data we produce nowadays.

A number of projects and working groups elaborated two major strategies to face this problem and to preserve digital objects over a longer period, namely Emulation and Migration, which can be subdivided into a wide array of possible solutions. Both strategies try to transfer the original digital object and its environment to a newer platform, but on different levels of abstraction. Emulation is taking the more radical approach, trying to simulate the whole computer environment of a file. Therefore all relevant characteristics, starting with the processor speed or the screen pixel resolution and covering many other aspects, have to be captured together with the file at the time of ingest. The second major preservation approach, Migration, considers the file itself. It is being migrated to the new environment by changing the file encoding in such a way that it can be reopened with future software programs.

Additionally, some alternative solutions are suggested ranging from computer museums (where computer machines with their original software are stored as long as possible), to output only machines with an independent energy supply and very stable components [21]. In order to provide an overview of the various preservation methods and to build a basis for the other chapters, these various preservation possibilities are described in detail in Chapter 2.

Many implementations of these strategies have been tested, rated and implemented, but until now, none of them is clearly better than all the others. Thus,

many inexperienced users, but also experts are not sure which solution to use for which data collection. Unfortunately, the decision depends not only on the size and composition of a collection, but also by many other qualities, such as user satisfaction or costs.

To support the decision process, a testbed is described in Chapter 3 which provides a framework for storing a wide array of files in many different formats, and for describing them with metadata. Whenever a preservation process is evaluated these files may be taken as a reference material, making changes in the files characteristics obvious with the help of metadata descriptions. This testbed is described in detail in this thesis in Chapter 3.

Choosing the optimal preservation strategy for a given setting constitutes a complex decision, depending on a large range of possibly conflicting requirements. In the area of research in infrastructure projects, complex decisions have long played a major role. Bridges, dams, and highways have to be built the best possible way, also requiring to obey many different constraints. Therefore *Utility Analysis* [49] was developed as a tool to integrate and evaluate very different aspects, to give an overview over them, and to accumulate them to a single decisive value. As presented in Chapter 4, Utility Analysis can also be used in the preservational context, which we will demonstrate at the same time theoretically and practically by evaluating preservation solutions for an editor's MS Word 2002 collection. As a second more complex example, the requirements of the Austrian Phonogrammarchiv concerning the preservation of its audio files were elaborated and are presented in Chapter 5.

2 Preservation strategies

2.1 Introduction

During the last couple of years, many ways of saving digital collections were presented. Many of them have been in use in various areas and tested in research projects all over the world. Nevertheless, there is still not much worry about the long-term preservation, although many files and data have already been lost. In this chapter, the most common ones and also some rather new preservation strategies are discussed, focusing on Migration and Emulation as the most promising strategies. Additional, upcoming and specialized strategies are described in a third part.

Also research focuses on two major preservation families, which are Migration and Emulation. The idea of *Migration* is to adapt the encoding of a file to the actual hard- and software platform, thus changing the electronic record periodically. *Emulation*, on the other hand, focuses on saving the record together with all necessary software, documentations, and specifications, in order to be able to re-open the digital object in later years. Both require initial and ongoing effort, but the likelihood to succeed is fairly high in contrast to the third strategy as described next. Together with *Migration*, *Standardisation* is presented, which is already a kind of Migration and which is often done before ingesting files into a preservation environment. Together with *Emulation*, the preparing strategy *Encapsulation* is described.

The third group, which is widely spread, has not been discussed in such a detail in the preservation communitiy. It is here called *Alternative Strategies*, a contains all other approaches, which are not covered with the first two chapters. Nevertheless, in order to give an overall view over the state of research, and to present all possible strategies for preserving media, also these methods will be described, discussed and evaluated. One of these alternatives is the printing of the whole data collection, which is a widely used method although it completely disregards the favours of an electronic record, other are the construction of a digital tablet or the long-term maintenance of computers in computer museums.

To sum up, this chapter consists of the following parts:

- Migration and Standardisation
- Emulation and Encapsulation
- Alternative Strategies

This chapter intends to give an overview over the various preservation strategies and discusses their strengths and weaknesses in order to facilitate the choice for one for future preservation projects.

2.2 Migration and Standardisation

As a first strategy, *Migration* is described. Since many migration solutions start with a standardisation process, this first step is described before going into detail on the *Migration* strategy itsself.

Standardisation

The idea desribed here is to preserve the digital object in one of the standard file formats which are widely available. By doing so, the effort for further strategies can be reduced by limitating the amount to a few commonly used data formats. This would even work as a strategy of its own over a shorter time horizon, because it is more likely that commonly used file formats of older platforms can be re-opened at newer hard- and software environments.

Either files are created in the necessary format, or they are migrated into it, for example from Microsoft Word to Adobe PDF. Better by far is the direct creation of a file in a standard file format, because it increases the probability that the original file will not be altered in its appearance and behaviour. Even information might get lost when converting a file into a suboptimal or different standard. For example animated illustrations, Macros or the underlining of written words get lost when transforming a Microsoft Word file to Adobe PDF. More details can be found in the next section.

Standards are widely applied, possibly open, and well specified. Open standards have two advantages. Firstly re-engineering will be easier due to the open specification, if they will not be supported any more one day. A second advantage is, that using these formats, no license fees have to be paid and no legal problems arise. On the other hand, proprietary standards are in some areas widely used. Examples are the MS PowerPoint or the MS Word format, and due to the monopoly in changing the format, they are under a better control and can be standardized more easily.

Although there are at present no 'de iure' agreements or guidelines on common standards, some 'de facto' standards may be derived from the fact that they are being applied in several important preservation projects. For example: In the Archaeology Data Service in York [2] a range of required standards formats is defined; in the pharmaceutical industry [10], TIFF, XML and PDF formats are used in some companies to store their data; or in the VERS project structure is saved with XML and content with PDF [47].

One has to be careful with standards when they have not been sufficiently

specified so they may change over time. This happens frequently when further features are being introduced or slightly different specifications come in use. An example are databases where 'Outer Join' can be used in some implementations and not in others. Other problems might occur from unused features when various projects preserve different subsets of this implementation. Another disadvantage is the limited usability of this strategy in many fields, such as digital art, where a change of the file format can alter the appearance as a whole significantly.

Although there is no general agreement on which standards to use, some formats are widely used and handled as standards, as can be seen in the following list. The list is aligned to the standards accepted by the Archaeology Data Service [2].

- **Text**: A simple ASCII string is accepted to safe the content, also PDF and RTF are very often used. Other often-supported text formats are HTML, LaTex, Postscript, SGML, TeX, Word or WordPerfect. It is very likely that UNICODE will get the same importance as ASCII, because of its ability to represent special characters of different languages. In addition, XML might become very important here, especially because the newer generation of word processors adopts XML as their native data storage mechanism [15]. Other elements embedded in the text, such as graphs and spreadsheets, are better saved separately. At the current time a common PDF-Archiving Standard [18] is under development, which may become the leading format for preserving documents.

- **Graphics**: TIFF is the accepted standard, others often used are BMP, GIF, JPEG, PhotoCD, PNG or CGM for vector based formats. Additionally some of the document standards, such as PDF, PS or EPS can be used for graphics as well.

- **Spreadsheets**: As with texts, also for spreadsheets a simple ASCII string is accepted, where data are separated by a semicolon or something similar (comma separated values). Other often-used standards are Excel, Lotus or Quattro Pro.

- **Audio and Video**: There is no general accepted standard for the preservation of audio and video, frequently used standards are MPEG, Quicktime, MIDI, AVI or WAVE or an extented version of WAVE, the Broadcast Wave Format (BWF), which is most often used for preserving audio data [44]. In the MAC/OS world, the Audio Interchange File Format is additionally often used [44].

- **Databases**: for databases containing more than a few tables, especially SQL is widely accepted, but also CSV files or XML-based interfaces.

- **Programming**: Programs in independent and open languages, such as JAVA, C, C++ or COBOL, are preferred to proprietary languages, such as Visual Basic or C#.
- **Presentations**: The question of how to preserve a presentation is not mentioned often. The most often used tool is Microsoft PowerPoint, with its PowerPoint format.

Reducing the range of file formats is a very rewarding process before going on with another preservation strategy, because of the far smaller amount of effort required afterwards. Since most files are already created in one of these standards, the preservation complexity can be far reduced by transforming only a small amount of files.

Migration

The most common preservation method is Migration. Most of the time it is applied without even recognising it. In addition, many preservation projects decide to use this method, because of its favourable price and because of the small initial effort required. An example, where Migration is already used is the pharmaceutical industry [10], most preservation projects, such as Camileon [50] or Cedars [6], also spend a significant share of their effort on Migration.

The basic idea of Migration is to update electronic records by transforming them from one hardware or software generation to another. Technically Migration can be divided in four groups depending on the amount of changes to be applied on the digital object [7]. Depending on the amount of changes, these possibilities are used for very different purposes.

- Refreshment
- Replication
- Repackaging
- Transformation

Only minimal changes are made with 'Refreshment', where only the bit stream is copied from hardware media to another without altering the bit and byte stream and without altering the storage-mapping infrastructure. This method is mostly used for copying CDs or disks and to avoid a storage media disruption. 'Replication' may require a change in the way the bits and bytes are saved, if one of the two storage media is organized. 'Repackaging' changes even more, here a part of the information of the encapsulated files in form of packaging attributes is altered, but the bits and bytes of the original electronic record remain untouched. Finally, 'Transformation' tries to preserve the content and appearance

of a medium but changes the underlying bit code of the digital object. Most of the time preservationists speak about Migration instead of using the detailed form of Refreshment, Replication, Repackaging or Transformation.

Also standardisation may be seen as a kind of transformation to improve the possibility of reopening the file, aiming in this case not so much at the version, but on the file format. The aim is rather to reduce the complexity of working with the digital object in order to improve its chances of surviving over a longer period.

An advantage of Migration is, that in many programs, it can be realized relatively simply with the help of backward compatibility. Users often do not even realize the transformation act. Backward compatibility makes sure that a new version of software is able to process files saved with an older version of the same software. Usually these electronic records will then be saved in the file format of the newer software version, which is a kind of Migration. The altered file can with some effort be compared to previous versions of the same file in order to control if any unwanted changes did occur in the new version. Migration gives a certain guarantee to the user that the digital object will be accessible within some years without any extra technical or preservational treatment [47].

There are some problems with Migration: Continuous new developments make it necessary to renew Migration approximately every five years [32]. Without automated processing, the effort to convert large data collections will be enormous. A second disadvantage occurs when new formats of programs are not backward compatible or when the underlying paradigm is completely changing. One example is the Migration of a relational database to an object-oriented one, another the conversion from Word to PDF in order to reduce the use of proprietary formats. In such cases, if bad comes to worse there is no way out at all and other preservation strategies are to be considered, or it may be necessary to write explicit Migration programs.

Another well-known problem using Migration is that due to minimal changes in the version of the format, information may change, such as digits in a float number, underlying layout descriptions or the functionality of certain calculations. What can then be done? Manual corrections may cause a lot of work while on the other hand the file may undergo alterations to an extend that it does not represent the original any more. Taking all this into account, chances are high that the Migration conservation method will afflict a file in some way or another over time and will loose its original look, feel and behaviour.

Jeff Rothenberg [42], an American preservation scientist who is himself strongly supporting Emulation, points out six problems in Migration: According to him, Migration is

- Labour intensive
- Time Consuming

- Expensive
- Error prone
- Risky
- Non-scalable

All of these disadvantages result from the necessity to apply Migration in circles, while next generation shifts may never be predicted without risk.

Migration is used mainly for large and homogeneous collections that run on standardized or partly automated processes. There is less risk as long as the look and feel of the file is not the key criterion, but the content; and also as long as the file does not include executable files, which may have to be re-programmed at every Migration step. Finally, it is easier to migrate between non-proprietary standards, so the code of the saved files is known and that Migration programs can be written eventually.

In literature five reasons are given, when to decide on Migration, some of which might also count for standardisation [23]:

- The file format is obsolete or has a declining market share.
- The actual format depends on certain hardware or software products with the risk of being abandoned or of being replaced by another standard.
- The format is proprietary and the vendor is not willing to publish the necessary information.
- A collection based on a wide range of formats, which are to be reduced.
- The need for metadata is increasing and more information can be embedded in the migrated file format.

Research on assessing the risk of Migration was made by the Council on Library and Information Resources [23].

There are four possible approaches of migrating a collection: One is to convert the collection to a standardised file format to reduce the maintenance complexity. Another way is to store files in their original bit stream and to migrate them only on demand, a method, which is called 'Migration on Demand'. The third option is 'Migration within the same format' in which the original bit stream is altered every time the underlying format changes significantly. Finally, another rather different approach is to migrate the software environment to a new platform instead of changing the electronic records. All four possibilities are described in the following paragraphs in more detail, and evaluated according to the dimensions of costs, risk of disruption and data loss.

Migrating to other formats

Expenditures....medium, *Risk of disruption*....high, *Risk of data loss*....medium

Due to the relative wide meaning of the word Migration and of possible strategies, which belong to this direction, different ways of migrating a file are described here. The first one is to change the underlying format of a file, e.g. to migrate it from MS Word to PDF to reduce the variety of file formats in a collection or to save the data on more reliable file formats. A second reason for choosing the 'Migration to other formats' alternative is, when several versions of a file are to be kept to reduce the risk of one file format not being readable any more. Problems and chances linked to this kind of migration are further described in Chapter 2.2 'Standardisation'.

Migration within the same format

Expenditures.....medium, *Risk of disruption*.....medium, *Risk of data loss*.....low

In contrast to the first poosibility files are migrated here whenever major changes in the underlying format occur. A good examples is MS Word, where files are migrated to a newer format encryption, whenever opened with a newer version of the software.

If a format turns out not to be legible with new software, the files are migrated to this new format family. This method decreases the frequency of Migration substantially, thus reducing also the effort and possible mistakes.

Mind that it can hardly be predicted when the next paradigm change will take place; also required work may not be forecasted; it is not required continuously, but in waves after version changes.

Migration on Demand

Expenditures.....low, *Risk of disruption*.....low, *Risk of data loss*.....medium

Migration on demand makes sense when the costs of continuously migrating files are either intolerable high, or when writing a viewer - a program that is able to open older versions of the same format - is possible. It may then be a good idea to store the collection in its original file format, planning transformation only if needed, only 'on demand'. One obvious advantage is the decrease in Migration costs as long as such viewers are available. Another positive aspect is that the risk of disrupting files with continuous format changes can be decreased as long as the original file is the basis.

The major problem may be the availability of viewers. They do not exist for many data formats and the gap to the original format might be too big to cope with, making the writing of an appropriate viewer impossible. Another disadvantage is the fact that more metadata for the files will be necessary to describe their origin and their behaviour than for the 'Migrating the whole collection'-strategy. Finally, viewers might not represent all original characteristics of a file.

A third Migration method, which combines the benefits of the two, previously presented relies on backward compatibility and is called Version Migration.

Software Migration

Expenditures.....high, *Risk of disruption*.....high, *Risk of data loss*.....medium

Finally, a slightly different alternative is the adaptation of the software as a whole to a different platform. This strategy is also called Re-Engineering. It is useful when there is a high dependency on a specific platform where no further versions are provided. In order to maintain the functionality of the software, the original code has to be altered to fit the new platform. This can be done by adjusting the program of the code, or by writing completely new software, which allows the copying of the functionality, or by translating the compiled instructions directly to the new platform with the help of a new conversion program.

The approach where a program has to be migrated requires high effort. Finding the original software code is often difficult if not impossible; and finally there will be complications with the users licence agreement with the software companies, because altering and recompiling the software code is usually strictly prohibited. Thus, Software Migration can be only used when the necessary rights have been legally granted and when expertise, tools and the source code are available [47].

2.3 Emulation and Encapsulation

As in the previous section, *Emulation* is decribed together with *Encapsulation*, which is done to prepare a file for long-term preservation. Again, this preparational method is presented first, and the *Emulation* itsself.

Encapsulation

When an object is preserved for unknown future platforms and users, far more information must be saved than just the original bit stream. Examples are software necessary to open a file, a description of the original appearance or the size of a file. This information is kept with metadata and saved together with the original object. This form of storage is called Encapsulation. The capsule contains all necessary information to reopen a file in the future. It can form a basis for Emulation, but also for Migration and as a short term stand-alone strategy: If files are in danger of getting lost and no other preservation strategy was prepared, the future users will with the help of the data contained in the capsule probably be able to reconstruct an appropriate platform to re-open the file in a legible way. That is the same with Emulation and Migration projects, where the encapsulated information can be used to define major differences to the original document.

A first question is what metadata are saved together with the original document. There are some different approaches of what to save; one well known is

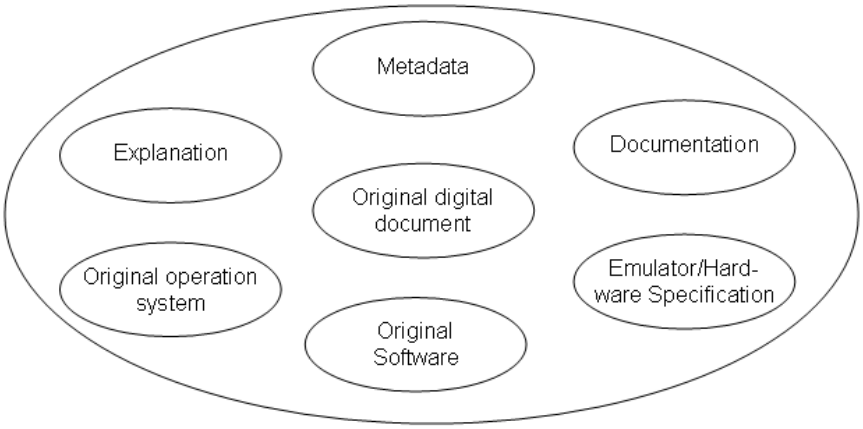

Figure 2.1: Encapsulation

the approach of Rothenberg [42] as can be seen in Figure 2.1.

The data blocks are either saved directly inside the capsule or linked to the same material with the help of a reference. Such data can be sub-divided in three groups [13]:

- Original data and software
- Specification for a future platform
- Metadata

The first group of encapsulated data consists of the preserved object itself, which cannot be seen without the original code for the software, which runs the file on the original platform, and the underlying operating system.

The second group collects all specifications for a future platform, on which the original software is able to work. Here all the relevant attributes of the original hardware environment must be described. They range from the execution speed to the display pixel size, from the calendar representation format to the peripheral characteristics [42].

The third group describes the file itself. It includes a documentation of the original object, describing its appearance, the number of pages and the functionality in natural language, but also the functionality of the surrounding software and hardware environment. These data are saved in a standard text format, which is migrated to a state-of-the-art standard in order to open it without problems. The addition of the used alphabet may be helpful, if special characters, such as in German, Chinese or Russian are used. Actually, XML is widely used to structure these metadata, because it offers the possibility to understand the plain code

without digital interpretation. Additionally to the description function, metadata are used for search operations. Sometimes metadata of a collection are bundled on a single server to speed up the searching process.

Other, but rather similar suggestions of what to store inside the capsule were made by the 'Reference Model for an Open Archival Information System (OAIS)' [7], by the OCLC/RLG Working Group on Preservation Metadata [29] or by the Victorian Electronic Record Strategy [48]; a fourth attempt in this area is the 'Dublin Core' framework, which enlists fifteen elements to describe the essential features of a document [8]. In contrast to the first three, the focus of Dublin Core is not preservation, but a common standard for the description of digital files.

When designing metadata, attention has to be paid to the future user. Here the following four principles are considered and should be kept in mind [9]:

- Modularity
- Extensibility
- Refinement
- Multilingualism

Modularity aims towards saving the Encapsulation modules separately and independently from one another in order to facilitate the re-usability of certain modules in other contexts. Extensibility means that it has to be possible to add specific metadata tags to basis metadata attributes, particularly if further specific description (such as a link target) is needed to describe a digital object. Refinement refers to two different aspects. One defines the amount of metadata used to describe the record. It has to be decided between a detailed but more expensive approach and a cheaper, but possibly insufficient solution. Refinement then allows to gradually add more metadata where needed. On the other hand, Refinement refers to the restriction of values for a certain metadata element. Such a limitation to a certain defined vocabulary or to certain allowed values facilitates automated processing. Finally, multilingualism focuses on the international use of electronic objects. It is recommended to work with metadata definitions, which may be easily translated, into other languages, or to use definitions that are standardised and available. The European Union, for example, is using the LOM specification for its metadata [9].

Additional objectives to keep in mind when designing metadata are [48]:

- Self-documentation: The metadata should be accesible directly only with the help of a simple text processor.
- Self-sufficiencency: The readability should be possible without being dependent on specific hardware or a specific computer environment.

- Content documentation: The format, which is used to encode the stored file, has to be described in detail.
- Organizational preservation: A description of a document‘s background to understand its context and content.

Implementing metadata, the preservator has to have an appropriate specification to create in the future an environment, which will allow the reopening of a digital object, or at least of the relevant attributes. Here the role of metadata is to verify if an object is presented in a correct way, whereas it is the role of the emulator specification to enable the reconstruction of the hardware environment in the future. This reopening will be discussed in more detail in the Emulation chapter.

Additional projects with a focus on metadata and encapsulation are the Text Encoding Initiative [5], where it is tried to describe the whole content of digital texts with metadata in XML. Others are the projects of the Research Library Group [39], where information for specific historic events is systematically collected and digitally preserved. A third project in this area is finally the PANDORA project [35], where the whole process of storing files is being investigated, also mentioning necessary metadata [4].

Emulation

Emulation focuses on the recreation of the hardware environment of software, as opposed to Migration where the file itself has to be adapted. The idea of Emulation is to apply software, which is able to simulate the behaviour of an older hardware surrounding. In this simulated area, old software can be run and old files can be opened in their original way.

Emulation consists of two steps. The first is the initial effort to prepare a file for preservation; the second is the creation of an appropriate emulator.

Preparing the files: The files are prepared by encapsulating the data in question. This capsule does not contain only the original software but also additional information, namely a specification for an , as described in the previous subsection. The emulator, which will be necessary to rebuild the environment in which the file may be opened.

Creation of an emulator: In order to be able to reopen the file in a new computer environment, a program has to be written which simulates the original environment. The necessary information is kept in the encapsualted file. The documentation of the original document is used to assure that the emulated file does not vary too much from its original in its significant properties. This emulator has to be designed once for the specific environment of the file to be opened, but can then be used for all possible files within the same environment. Emulators are already widely in use for example in the computer game community [40], or when old software has to be run on a newer hardware environment until also

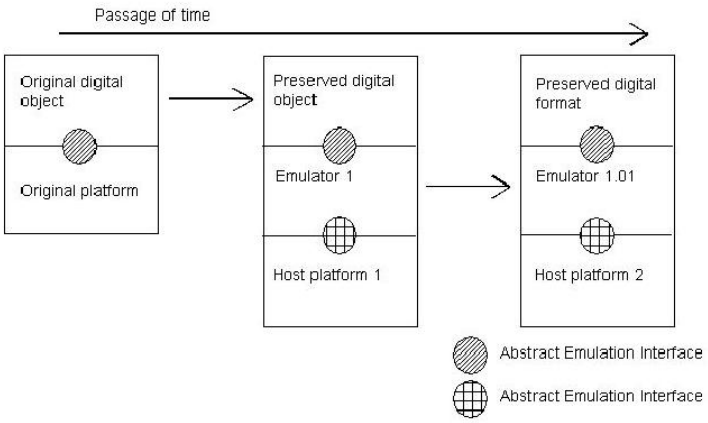

Figure 2.2: Emulation [50]

the software is updated [12]. A specific example is the Multiple Arcade Machine Emulator (MAME) [43], a program that is able to emulate a large number of different platforms at the same time.

As can be seen in Figure 2.2, a new emulator has to be written, whenever the underlying platform changes. The preserved file itself remains untouched. If the new emulator has the task to link the new environment with very old emulated hardware, in theory it would do to emulate into the previous platform only and to run the older emulators inside this new program. This would reduce the whole step required for a new platform to the writing of a single new Emulator. But because this second approach results in a very high complexity, the first approach of writing new emulators for necessary previous platforms every time the platform changes, is usually preferred.

A second reason for not emulating emulators is that the effort necessary to create the emulator itself is minimal in comparison with the effort to create the interfaces between the digital object and the emulator and between the host platform and the emulator. Adding a second emulator would require to write another new interface, probably more complex than the interface of the preserved digital file.

The biggest advantage of Emulation lies in the fact that the preserved file itself remains untouched without modification and so cannot be corrupted or altered at any time or in any way. With the help of a good emulator, the appearance and behaviour can be completely simulated to the extend that there is no differnce to the original. In fact, what can be seen is the original whose presentation becomes

possibly through an emulator. As a solution, metadata and documentation are necessary to describe the original object in more detail.

Another advantage is that emulators have been well introduced for many years. In the computer industry, they are in use to simulate the behaviour of a future computer platform. The computer game community emulates to allow the playing of older computer games in new hardware. Here the users will ask themselves whether they find differences to the original game, or not. According to demand in industry and communities, many emulators already exist, and not a few people are experienced in creating new emulators.

Emulation is a fairly complex problem, though, and the amount of future complexity cannot be predicted, so it might become impossible to create emulators after a platform paradigm shift. Furthermore, no standards for preservation and documentation have been defined, so efficient emulation might have to cope with inadequate documentation. Another problem will come up when future users will not know any more how to interact with the programs that are used today, so there will be demand of adapted, migrated software to handle old files.

To sum up, Emulation is used when complex data seem sensitive to changes, when enough capacity for encapsulating the objects is available, and when expertise for writing emulators exists.

The Universal Virtual Computer (UVC)

In the UVC-Approach a virtual platform is created which is linked to other real computer environments with special emulators. From this platform, emulators to other real computer environments will be available. This virtual platform covers all necessary attributes and on the other hand have clear and easy interfaces. The idea is that standardized emulators from actual platforms to the virtual computer can be implemented with far less effort than if a program for a real platform has to be written specifically. A UVC specification is under development, some parts have already been published by IBM [24, 25]. A first implementation has been introduced at the Koninklijke Bibliotheek in The Hague [31].

2.4 Alternative Strategies

Technology Preservation

If no preservation strategy is at hand, data can be kept simply by preserving the underlying hardware and software. So, by running the old equipment, digital objects may be in use for another five to ten years after this environment goes out of the market.

A second possibility is to store only the digital objects themselves, relying on computer museums or computer societies to keep the hardware environment running. This approach is a good choice if no other strategy is at hand and if

there is danger that data are lost. At least one advantage exists, that data are presented with their original look, feel and behaviour.

However, there are also some important disadvantages. It is due to chemical disruption very unlikely that old machines will run indefinitely. A second problem is the transport of modernly stored files to these old computers. In order to transfer the data, a new interface for modern storage media has to be produced for the old computer, requiring knowledge, which probably does not exist any more.

However, there are two other good arguments for the installation of computer museums: In some cases computer museums may be the only possible solution, e.g. for cars with electronic equipment, airplanes, ... The second is, that when migrating digital data, it shall be never again able to restore the original version except with the help of the original technique. Even comparison is impossible without the natural presentation. Old computers are therefore to be maintained for scientific reasons - at least as long as this is possible at reasonable costs.

The Digital Tablet

An uncommon approach to computer museums, made by Kranch [21], is to develop a 'digital tablet', which displays information and which is equipped with a touch screen. The information shall be altered and viewed on a screen. It is planed that the machine contains an autonomous energy supply and is able to survive thousands of years in neglect under harsh conditions. Finally, it contains a read-only port as well as a description how to download data out of the machine. Because of a number of unsolved problems, such as the energy supply or very durable hardware, this strategy is not widely discussed yet.

Considering the case, that not even this strategy can be afforded, or that digital objects were not preserved at all but are found later and are read again, data recovery is the last possibility.

Data Recovery

With this strategy, it is tried to recover obsolete or damaged digital objects, to reconstruct the bits and bytes stream and to interpret them in a correct way. The process of recovering such files starts with the recreation of the bit stream. Means are microscopes to read magnetic tracks or chemical treatment of the corrupted data storage [41]. The next step is to locate certain patterns and indicators in order to detect the file format and the original interpreting software. This can be often very difficult without the existence of appropriate metadata.

Recovery is necessary, if data get lost. It can happen when their format is not supported or when the storage medium cannot be accessed any more. This happens due to too old storage media and due to a disruption during the regular life cycle, which might be a result of one of the problems presented in Figure 2.3,

taken from [41]. Note, that this figure does not include the risk of format obsolence at all.

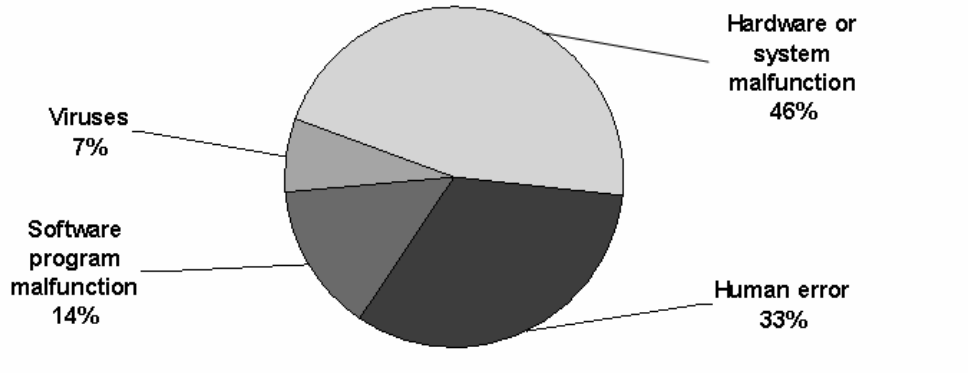

Figure 2.3: Reasons for the loss of data

Because of these reasons, data recovery is an insecure alternative and the preservation cannot be guaranteed. A helpful means is to provide metadata, so some opening obstacles may be minimized and it is easier to reconstruct the original look and feel. A way of reducing the risk of the second, third and fourth obstacle in Figure 2.3 is the use of a computer museum, where the original software will be available and can be used as long as the preserved hardware works.

Data recovery is often very expensive, requiring new efforts for every single data object, implying that the required effort is hard to predict. Without proper documentation, only little can be done.

Printing

The idea of this last presented alternative is to print the preserved material and to eliminate the influence of digital technology through classical storage on paper, microfilm, or nickel plates. On nickel plates sold by Norsam Technologies with the size of 5 x 5 x 0, 6 cm, about 18.000 pages can be saved, if they are reopened with an optical magnifier and about 196.000 pages, if reopened with an electronic microscope [28]. The company claims, that the data persist for at least 1.000 years.

The second possibility is to print or copy material on permanent paper, as defined in the ANSI/NISO Z39.48-1992(R1997) norm, where requirements for the PH, the tear resistance, and the concentration of Alkalin and Lignin are defined [27]. But storing the data of a standard 2.400-foot magnetic tape by printing them out would need up to 135 cubic feet as storage area and more than 50 times the preservation costs, e.g. approximately $ 1.350 in comparison with $ 25 per year [22]. Finally, also microfiches can be used, which have a lifespan of up to 500 years [33], if they are correctly produced and stored. The strategy

can even be used for photographs, which can be stored well on a polyester base, whereas nitrate and acetate films deteriorate faster.

In addition, audiovisual material and motion pictures can be saved in a non-electronic way over long periods, although the process in these cases is more complex. Generally it is useful to create duplications for public use and to store the original files in a save environment.

The question is, whether saving data in an analogue or digital way is a better solution. Benefits of the digital approach are, that the objects are easier to search, copy and transmit and that some digital information cannot be printed and would so get lost. Examples are animation, software functionality, or a moving picture.

On the other hand, if the above-mentioned attributes are not of high importance, the saving of printed versions has the advantage that it can be read with human eyes or with a magnifying glass completely independent from soft- and hardware. So neither viruses nor an electricity crash can destroy the collection, it can even persist without human and computational care for years and centuries under the right circumstances. So due to small faith in computer technology many information and library scientists see the non-digital approach as the only possibility to store the data for hundreds and thousands of years.

Many companies [11] prefer the non-digital approach too to save their most important documents over a longer period. Due to the relatively small amount of saved data, and focussing on the content, this approach can be appropriate in this context.

On the other hand more and more files are being created digitally these days. Keeping in mind that many collections are planning to save data in their original look, feel and environment, the trend goes towards electronic preservation.

2.5 Summary

In this chapter the most promising, but also some lesser known preservation strategies were presented and discussed in detail. Additional to the two main directions Migration and Emulation, the two preparative strategies Encapsulation and Standardisation are described, which are often used as a preparation for other strategies. Furthermore alternative strategies, such as the printing of files and yet unlikely solutions, such as the Digital Tablet are mentioned.

The chapter provides the necessary foundations for the following chapters, where specific parts of preservation research are mentioned - always related to the here presented strategies.

3 The File Format Test Bed

3.1 Introduction

In this third chapter, a File Format Test Bed is described. It is a framework comprising files of a wide range of formats to facilitate the test of preservation solutions. After the decision to preserve a collection, a user should, as a next step, evaluate several implemented preservation solutions, such as migration software, emulators, or other preservation possibilities. In order to evaluate these alternatives and to detect even little errors, this test bed provides a wide collection of files with various characteristics which are described by detailled metadata. These data display their characteristics in such a clear way that changes are visible right away. There are some legal difficulties with providing files which were not self-created material, therefore not many files are already available. Users are invited to add files as long as no conflicts with the copyright are to be expected, such as self-created ones.

This chapter deals with the idea, realization and usability of a test bed website. In Section 3.2, different testbeds are described and compared to the presented concept. Section 3.3 describes optical ideas and the website's functionality and also both the realization of the MySQL database and the concept of the PHP website are presented in detail in this section. The user interface design and implementation is shown in Section 3.4. The final output can be seen in Section 3.5, where the handling of the website is presented. Various functionalities and user options are explained in the form of a user manual.

3.2 Related work

There already exist some comparable testbeds, but most of them focus on file formats and not on the files themselves. One example is the PRONOM testbed from 'The national archives' in the Netherlands [34], where it is tried to define the file format and all surrounding criteria, which influence a format's behaviour. Another example is the DIFFUSE project where the focus is set on specifying all commonly used file formats [15], a similar community approach is taken at the 'MyFileFormats' project, where everybody is invited to post file formats and detailed descriptions of them [26]. Until now around 1.000 formats have been submitted. But in contrast to the here developed approach, these projects only store specifications of a format, but not representative files.

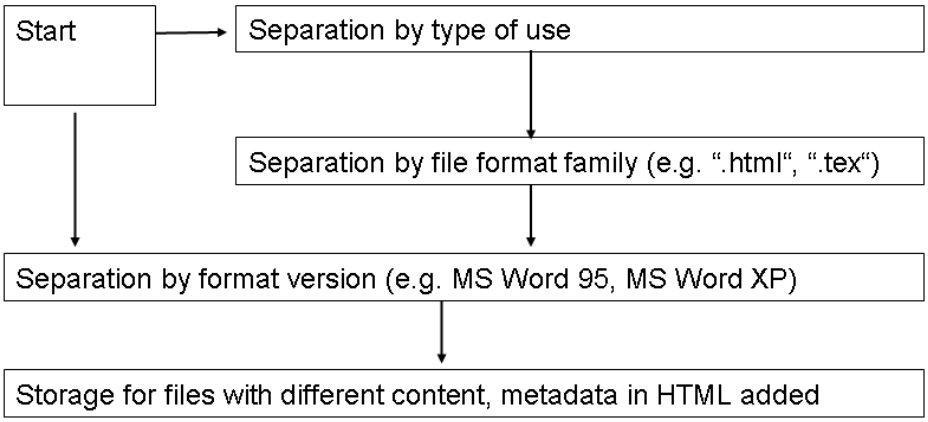

Figure 3.1: Website hierarchy

3.3 Database design and realization

The purpose of the File Format Test Bed website is to clearly arrange a large number of files in a hierarchical structure. In order to keep the amount of choices at each level within a limit of approximately fifteen, a so called deep hierarchy is used. As can be seen in Figure 3.1, this hierarchy has four levels, where the first and the second one can be accessed right from the start.

The first gives a choice between various types of use such as 'Documents', 'Spread Sheets', 'Databases', 'Sound', 'Moving Pictures', 'Graphics', 'Software', 'Presentations' and 'Others'. Within this first level, the different file format families such as Word or LaTeX can be found, according to their type of use. It is also possible to enter a file format list right from the first level. As a next step, different versions of the file formats can be selected, such as Word 95, Word 2000, and Word XP. Finally, the files themselves can be found and downloaded. The files are described with metadata which capture the important and typical criteria.

In order to introduce the hierarchical framework in a database, the next step is to create an Entity Relationship diagram containing different tables and their specific attributes. As can be seen in Figure 3.2, the four levels of the hierarchy are transformed into data tables:

Type table: Attributes are the species key and a species name. The species key is primary, the species name must not count more than 30 characters.

Family table: Its attributes are the type key, the family key and a family name. The family key consists of 3 digits which is the primary key of this table.

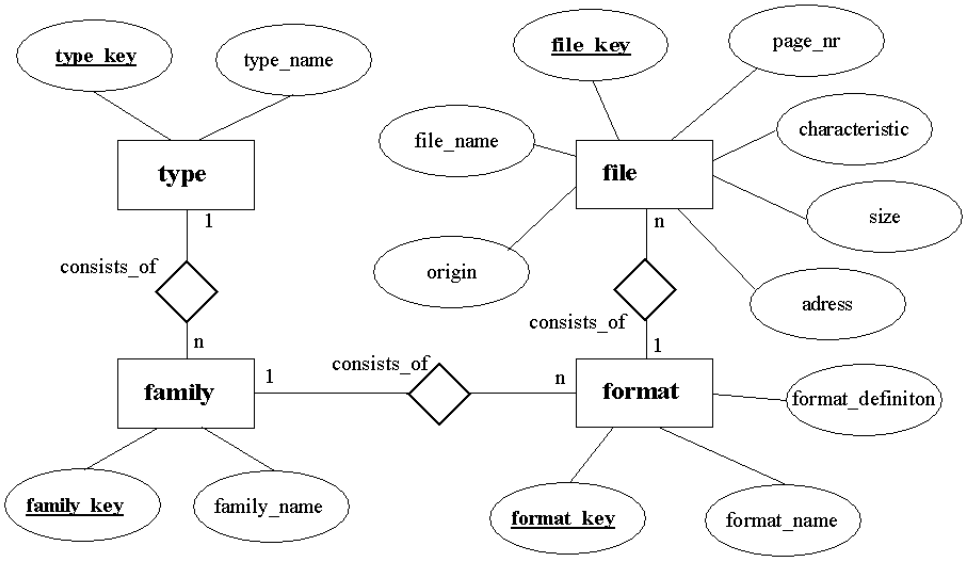

Figure 3.2: Extended Entity Relationship Diagram

Format table is slightly more complex than the first two tables. Again, a format key is used for identification. The relation with the species is stored in the family table. Additional to the two primary key attributes, the format table contains a format name, a format definition and an URL to a file format definition website. Such a website can either be the website of the owner of a proprietary file, or an independent definition. Many open source formats have been well specified and are already publicly available; an example is the very detailed specification of ADOBE PDF, which can be found in [1] another one is the specification of the possible ISO-PDF-Archiving standard PDF-A [18].

File table: This table is the biggest one, because the descriptions of the files are stored here. Again, the primary key consists of a file key. Additional attributes in this table are the file name, the origin, the file size, special characteristics and the URL of the file, indicating from where the file can be downloaded. The file name can count up to 50 characters and should use a descriptive name. The metadata standards, which are presented in Chapter 2, are not applied here, because of their strong focus on the history and context of a file - attributes, which are of no importance here. On the other hand, there are no official standards describing the appearance and structure of a file. Additionally, the significant attributes vary so widely, that a standardized description would not improve the description's quality, but only increase the effort needed to ingest a file. The here used limitation to only one 'characteristics field' is used, since no clear file appearance and behaviour description metadata set has been widely accepted. In the 'origin field', fill in the creator or the website, where the file comes from. The 'size field' has kb as the unit of measurement. The field characteristic is the

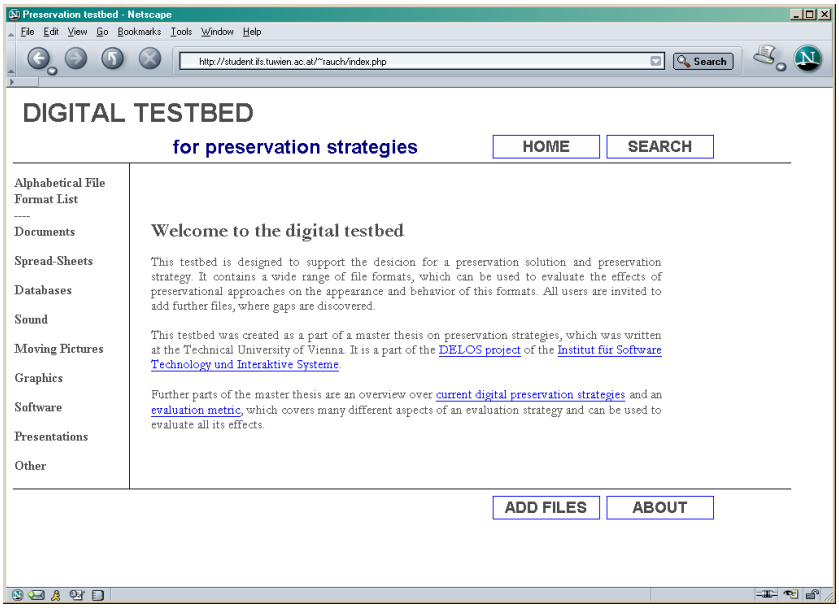

Figure 3.3: Website User Interface Design

most important one, because all the relevant information for describing a file is stored here.

There exist at the current time no conventions about possible criteria of a file and so the typical characteristics must be described in free words. They enable the user to notice possible changes, which may have occurred during the preservation process. In the fourth chapter, 'Choosing the right strategy', a detailed description of possible criteria of a file will be presented; it would be a good improvement of the website, to check all specific attributes with the help of a criteria list and possibly automatically instead of using free words.

3.4 User interface design and implementation

With the user interface the content of the databases is displayed in a clear and understandable way. Additional functionalities and content are required, such as a 'search' function and an 'add' function as well as a link to the start page and an impressum. Bundling all these requirements has led to the design of the website which can be found in the Internet at http://www.ifs.tuwien.ac.at/~rauch [36] and which can be seen in Figure 3.3.

On the left side of the website is the navigation bar. This is the starting point for finding a specific file. The search can start either by choosing the alphabetical file format list or by clicking on one of the nine different file species. These species cover a certain amount of formats. Because of the lack of a widely accepted file format classification, I suggest the following subdivision: Documents, Spread-

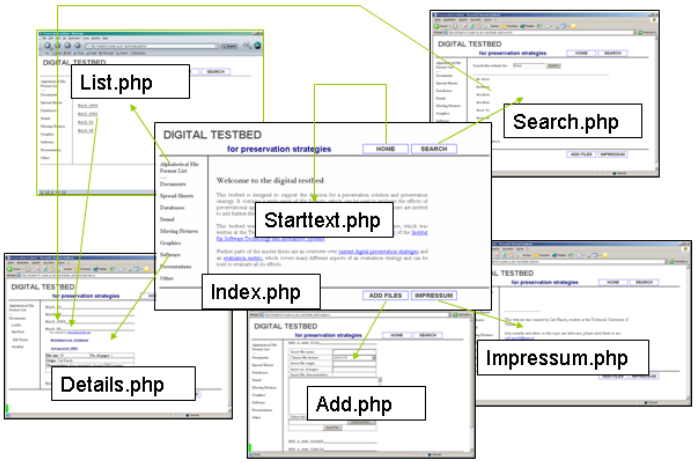

Figure 3.4: PHP architecture of the website

Sheets, Databases, Sound, Moving Pictures, Graphics, Software and Presentation. The different states and ways to use the main area can be found in the 'User Manual' Section.

The head line and the foot line of the page show the four buttons 'Home', 'Search, 'Add files' and 'About'. These system functionalities are completely independent from the information area of the website - the field in the middle on the right side - and can be started at any time.

The website is realized with PHP. To keep the PHP files at a reasonable size and in order to allow a certain modularity, the website exists of seven files. The start point is the 'Index.php' file, which constructs the headline, the left panel, and the foot line and which is visible during all statuses of the website. At the beginning, the 'Starttext.php' file is shown in the centre, but switches depending on the user's choice. The exact relationship between the files can be seen in Figure 3.4.

All files except 'Starttext.php' and 'Impressum.php' are connected to the database, although only 'Add.php' can write to it. To protect the site from unwanted entries, the 'Add.php' file is protected by a password. This secure access is realized by storing the file on an external secure server, where a password is required for every contact, and so also for this PHP file.

The most complicated part of the website was to create a clear overview over the stored files. As mentioned in the previous section, a four level hierarchy is planned and is here transformed into a visible output. During the design phase the aim was to make all higher levels of a stage visible and to allow fast switching

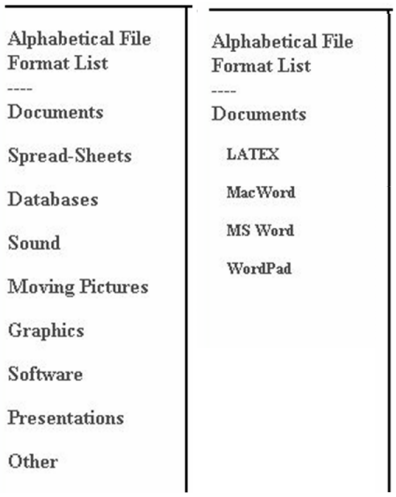

Figure 3.5: Hierarchy of the left control panel

between the different levels of detail. In Figure 3.5 the first part of the hierarchy is shown. The left part of the figure shows the initial appearance at the start page. This changes to offer the choice which can be seen in the central column as soon as one of the file species is chosen.

After choosing a file format family, the display changes as can be seen in Figure 3.6. By clicking one of them, all the files show up which have been saved in the chosen format. Finally, if one of these files is selected, a table with all the saved attributes will appear. Beside the file size, the file's origin and a text area for the explanation of characteristics goes along with the file. These output fields correspond to the fields in the database, as described in Section 3.3. There is a link 'Download the file'.

Beside the main part of the website to provide files, four additional features are added to facilitate the website's handling. The first one 'Home' is a simple pointer to the start page and leads users back if they lost their way in the data collection.

The second is the 'Search' function, where words or parts of words can be searched. The search area includes the format species, format families, file formats and the files themselves. According to their part inside the hierarchy, the results are printed big - for species - or small - for families, formats and files. When one result is chosen, the according page in the standard download environment will be opened.

In the 'Add files' page, either a file, a file format or a whole format family can be added and written into the database. For a further project, these functions can be extended by allowing to modify and delete the entries. Another improvement would be to define a list of important criteria which are typical of every file and

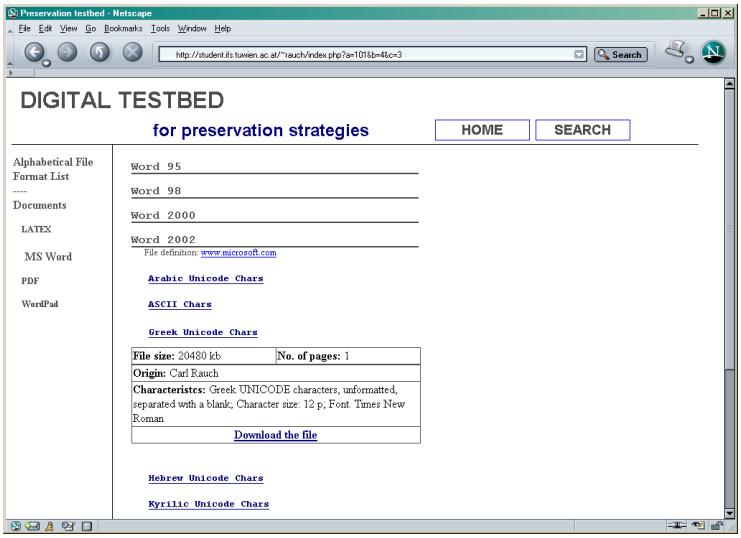

Figure 3.6: Hierarchy of the center frame

thus to facilitate the description process. So the individual description of file characteristics can be speeded up with some choice boxes.

The last additional functionality is the 'About' part, where some additional information on file format repositories can be found.

3.5 User Manual

The final part of the Test Bed chapter is a user manual, which gives some explanations on how to use the website. The chapter is divided into two parts: in the first subsection the functionalities for ordinary users are presented. They include for example the start page, the application of the hierarchy, or the search function. The 'Add' function is described separately in a second subsection because it can only be used by website administrators or authorized users. Therefore the Add page is protected by a password, which may be requested from the author by everyone who likes to add useful material to the website.

User

After typing the `http://student.ifs.tuwien.ac.at/~rauch` into the browsers, the start page appears as in Figure 3.3. Depending on the user's background and intention, one has the choice between searching for a file or exploring the additional functions. Because most users probably look for files, three ways to find the them will be described first:

Use of the hierarchic order: At the left side of the start page, there is a list of different species of file formats, such as Documents or Spread Sheets. In order

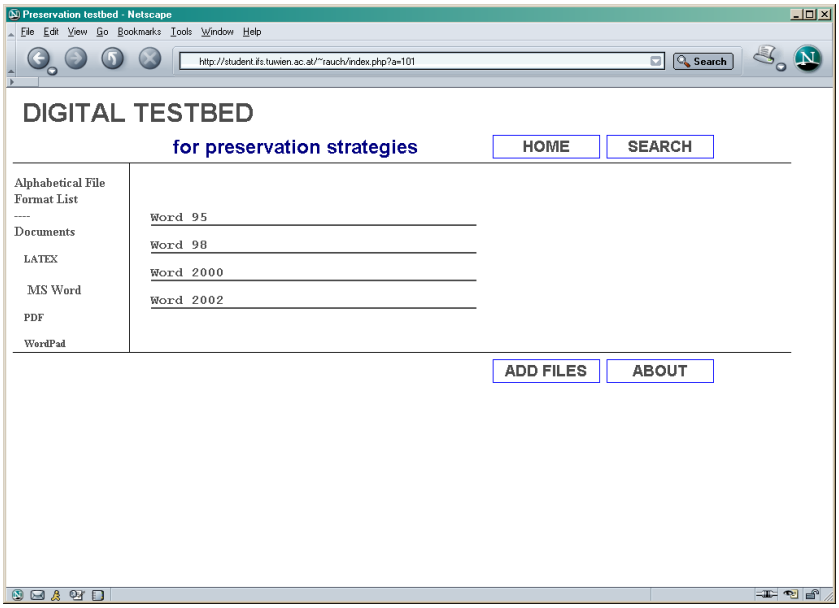

Figure 3.7: Choice between various file formats

to find an appropriate file, one has to click through the hierarchical levels. The first three steps are shown in Figure 3.5, after them a variety of formats appears in the center of the page as in Figure 3.7.

After choosing one format, all files which are saved in that format, will be enlisted. Taking one of them, a table appears, similar to that in Figure 3.6. This table contains the following information:

- **File size**: This figure is published here to enable the user to estimate the download time.
- **Origin**: To give a reference to the original context.
- **Characteristics**: This is the most important description field. Here all criteria are listed, which describe the differences to other files and which enable the user to detect changes through a preservation process.
- **Download the file**: Finally, the file itself can be downloaded.

Enlistment of all file formats: The second possibility to find a file is to click on the 'Alphabetical File Format List' link on the left side above the file species. This leads to a page which lists all file formats in the database. An example can be found in Figure 3.8.

If a file is chosen here, it appears in its hierarchical environment as in the alternative before.

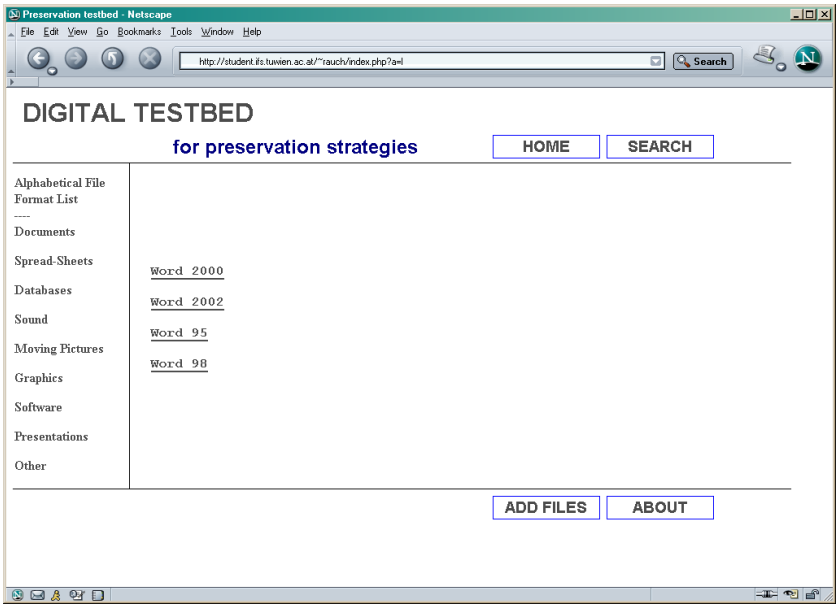

Figure 3.8: Choice between various file formats

Use the search function: The third way to find a file is with the search feature. Clicking on the 'Search' button on the start page, the user opens a window where he is asked to enter a line in a textfield. There is no minimal length requirement, so it is even possible to search for a single character. After using the 'Search' button, all entries of the database appear if they contain at any position the inserted string. The output is sorted according to the database in which the search result is stored, starting with format species which are printed in bigger characters, over file format families, file formats and finally the files. On a specific level the files are sorted alphabetically. Such a search result is shown in Figure 3.9.

Beside the possibility to search for a file two buttons are left, which have not been described. The first is the 'Home' button. As can be seen on the screenshots, the frame of the website is always the same and always visible. In case a user wants to return to the start page, the 'Home' button can be clicked.

The second button 'About' links to the 'Impressum', where the author, his e-mail address and the institution, where he is writing this thesis and as a part of it this website, are published.

Administrator

In addition to the user's functionalities, an administrator is allowed to add files, formats and format families to the test bed. Therefore the 'Add Files' function can be used only with a password. After clicking on the 'Add Files' button, the

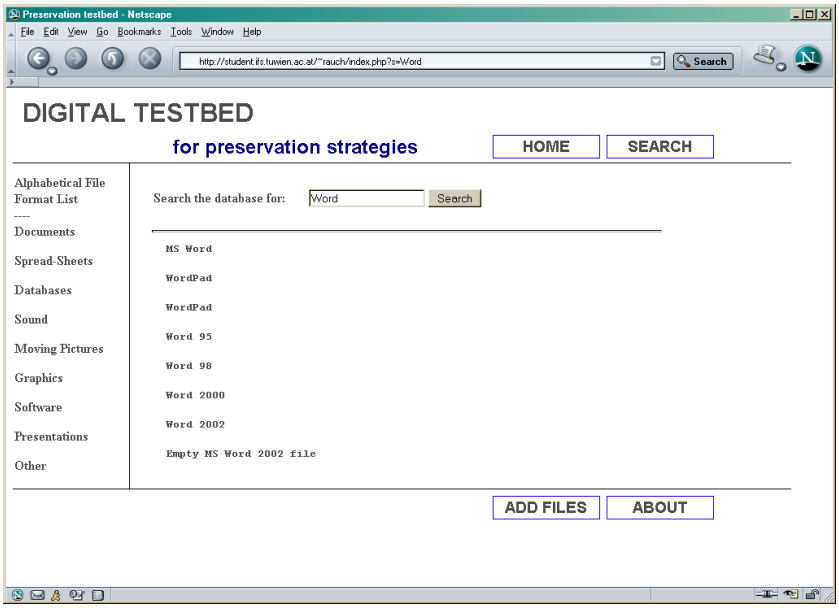

Figure 3.9: Result of the search 'Word'

administrator can choose if he wants to add a format family, a format, or a single file, as can be seen in Figure 3.10.

The interfaces behind the 'Add a new format' and 'Add a new family' are rather similar. In both cases, the name of the new entry has to be entered. This is the only obligatory entry, although all other fields should be filled as well. The second choice in the 'Add a new family' interface is to choose the format species with the help of the choice box. After pressing the button 'Add family', the new entry is inserted into the database. The differences to 'Add a new format', which is shown in Figure 3.11, are the following:

- Choice of the format family: Instead of the family species, the format family has to be chosen. Because of the supposedly high amount of families, they appear in alphabetical order.

- Insert definition location: Some file formats are well documented, such as PDF [1], here the user can find additional information, migration tools or older versions of a software. If no detailed file definition is available, at least the main producer of the software is referenced.

The third adding functionality is finally to include a new file. As can be seen in Figure 3.12, the following fields have to be filled:

- Insert file name: This field is obligatory. The file name gives an idea of the file's content.

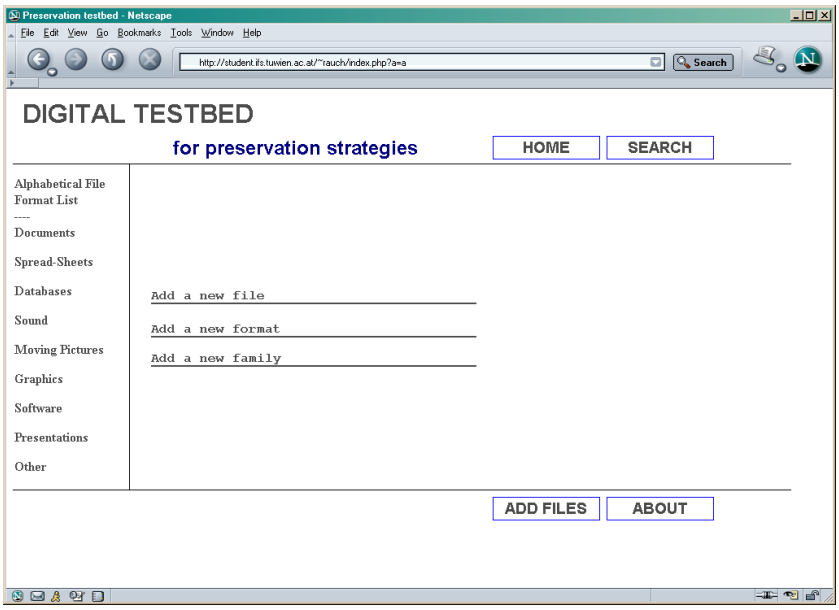

Figure 3.10: The start page of the 'Add Files' functionality

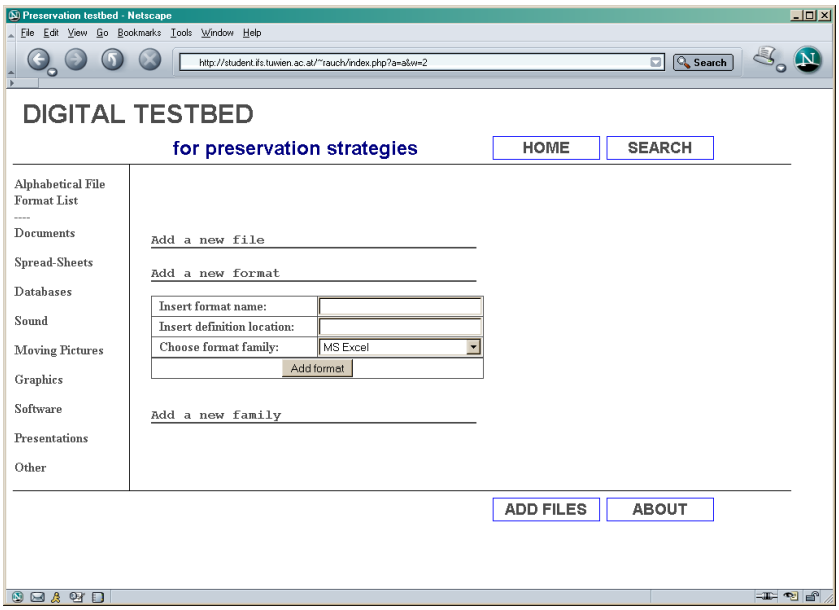

Figure 3.11: The user interface for adding a new file format

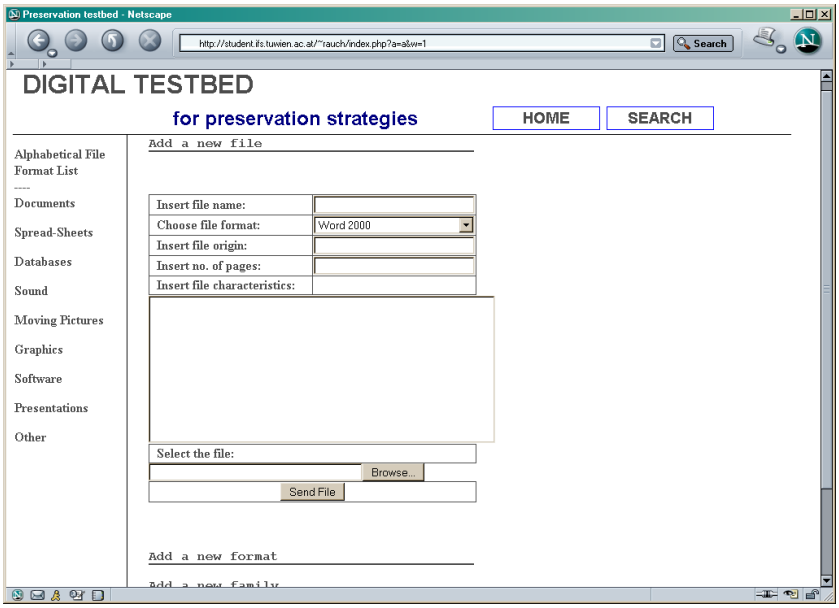

Figure 3.12: The user interface to add a new file

- Choose file format: Select the higher-level file format in the choice box. The formats are sorted alphabetically even though it might not always be easy to find what you want when the list is getting longer.
- Insert file origin: The website, from which the file is taken, or the author/creator is mentioned in this field. His name or the institution may help the user to find additional information on a particular file and on its background. Naturally, files are only published with the approval of the here referenced institution or author.
- Insert number of pages: For documents the number of pages can be inserted here. When reopening the file, this information can be used to verify, wheter the file differs significantly from its original.
- Insert file characteristics: This is probably the most important field. Here the administrator describes typical characteristics of the files which are saved. With the help of this field, the user can detect differences from the original appearance, structure and behaviour of a file. Attributes which can be useful here spread from a wide variety and can range from the letter size via the execution speed to the functionality of macros. A more detailed approach to classify file characteristics can be found in Chapter 4 in the creation of the objective tree. Note that not all criteria have to be inserted here, but those which are special for this file.
- Select the file: Finally the file itself can be selected in the upload field.

Clicking on the 'Browse' button, a file directory browser is opened and the file can be selected.

- Upload File: The last step is to push the 'Send File' button to insert the previously entered data into the database. After the click comes a window which repeats all inserted values. Reconfirm with 'File inserted', and the file is available in the website's structure.

After the presentation of the File Format Test Bed, where a user wanting to evaluate several preservation solutions can download representative test files with metadata descriptions, the next chapter describes a process to rank these alternatives. Here these solutions will be weighted according to their influence on the file's appearance, structure and behaviour, but as well according to process characteristics and arising costs.

3.6 Summary

Although only a few examplary files are inserted into the testbed yet, its functionality and its purpose are visible. In addition with other already available testbeds [34, 15], describing mainly file formats and file format specifications, the here presented testbed can be used to download files encoded in these formats. In order to evaluate a preservation solution, these files can be downloaded, ingested into the test environment and the outcome compared with the metadata, describing the significant attributes of this file. Therefore the stored data display their characteristics in such a clear way, that even little changes are visible right away.

In this master thesis, the first step - the creation of a framework - was done. In order to achieve full functionality, more files need to be added additional to these few exemplary ones.

4 Choosing the right strategy

4.1 Introduction

Utility Analysis has its origin in the evaluation and ranking of infrastructure and governmental projects, where it is traditionally in use for dam projects, for the planning of entire neighbourhoods, or for the evaluation of new transportation systems. Early scientific research in this area started around 1970, the here presented version was introduced by Arnim Bechmann in 1978 [3].

In English language literature the concept is referred to as 'cost-effectiveness', 'value-benefit', 'multicriteria' or 'benefit-value analysis' [49], but "Utility Analysis" seems to be the best translation.

For application in a preservation setting, Utility Analysis has to be slightly altered, but equals in most parts the original process. The workflow consists of eight steps shown in Figure 4.1, which are being described and discussed in this master thesis. The letters *"A"* and *"U"* in the figure specify whether a step has to be done by an administrator or a software system that moderates the process (A), or by a user (U), who has to make individual decisions. Steps two and five are done by either the user or the administrator. Here new values have to be entered manually, but the already known inputs can be re-entered automatically, wanting the user to merely control or adapt. The last step, deciding for the best alternative, is done by the user, although a numerical support and an initial ranking is provided by the system.

In order to demonstrate the usability of Utility Analysis, an example is given in the following subchapters, taking a file collection created with MS Word 2002. Additionally, another implementation of an objective tree for one specific scenario is shown in Chapter 5, which was created in collaboration with the Austrian Phonogramm-Archiv ('Österreichisches Phonogrammarchiv').

Related work

The research area from which contributions have been taken is economic evaluation. Economists use Utility Analysis when ranking alternatives for complex infrastructure projects. The combination with preservation questions constitutes a new application scenario, sharing many of the complexity characteristics. A good introduction to the traditional Utility Analysis was published by Hanusch [16]. He describes several predecessors of the concept as well as different analytical steps which have to be followed to receive a final ranking. A software to support

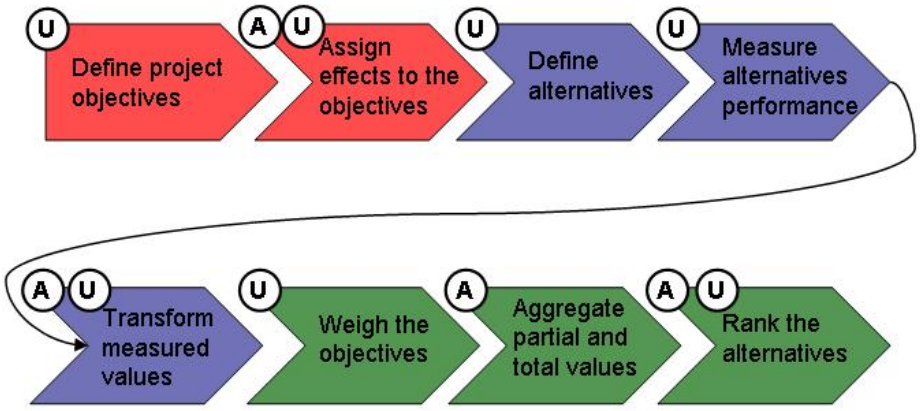

Figure 4.1: The eight steps of the Utility Analysis

the decision process was implemented by the Institute of Public Finance and Infrastructure Policy at the Technical University of Vienna [20].

In economic research Utility Analysis is often mentioned together with two other decision instruments [16]. The first one is *Cost-Effectiveness Analysis*, which focuses strongly on the representation of monetary costs. There are some scenarios where Cost-Effectiveness Analysis is not the first choice, weaknesses being that the status-quo alternative is not being considered, that the weighing metric is not as developed as in Utility Analysis, and that no clear ranking of alternatives can be done. The other model, *Cost-Utility Analysis* measures scenarios with monetary units. On the one hand this simplifies the comparison process, but on the other hand it reduces the explanatory power of the attributes and the transparency of the decision and requires a lot of effort and a level of uncertainty when determining costs of attributes like 'Font size' or 'Sound resolution' or when defining future costs. Concerning these disadvantages, Utility Analysis is probably the best choice for a preservation setting, providing, by the way, one very often used tool also in the infrastructure sector.

Some work has been done in comparing preservation solutions for different scenarios. The Research Library Group [17] for example migrated and emulated a software and users were asked to detect differences and to evaluate both solutions.

The work done in this chapter led to a publication at the International Conference of Asian Digital Libraries in Shanghai [37].

4.2 Defining the project objectives

The first step in Utility Analysis is to define the project as a whole and its goals, for example the file characteristics to be preserved. This is made by constructing

a so-called *objective tree*, where many different goals, high-level and detailed ones, are collected and put in relation to each other.

Generally, there are two ways to define goals, the *bottom-up* and the *top-down* approaches. The bottom up approach is to collect a list of basic attributes (such as font colour representation or staff expenditures), and to accumulate them on a higher level (such as the preservation of the look and the accountability of costs).

The opposite is done in the top-down approach where general aims (such as readability or cost structure) are defined first and separated into further detail in a next step. A synthesis of these two approaches is probably the best solution, combining high-level aims with basic requirements. Further on, two objective hierarchies are presented. In the first one, a tree is proposed as a high-level structure for future usage, the second one shows a practical implementation which is part of the example presented in this section.

The objective tree

The main goal in preservation research is to maintain access to digitally stored files in the future by preserving all relevant characteristics with respect to the given application domain at reasonable costs.

This main goal can be divided into the following two major groups: The first one consists of all characteristics concerning the digital object to be preserved. This object can be either a digital file, such as a MS Word Document, a whole software package, such as MS Word or even an operating system, for example MS Windows XP. All criteria necessary to describe this wide array of objects are collected in this first group.

The second group covers the preservation process. Here, all characteristics are bundled which do not directly refer to the preserved object but to all surrounding attributes. The outer border is the computer system which is used to execute the preservation solution. The separation is shown in Figure 4.2, where the inner part is the digital object, the white part is the preservation process, and around there is a surrounding system, which is the same for all preservation processes and whose behaviour cannot be predicted, so it is not considered in the metric.

One very important process characteristic are the costs. It seems advantageous to list them as a separate top-level goal because of the following reasons:

- The importance and direct influence of costs on many decisions might be overseen if they are listed as second or third level goals. When listed top-level it is possible to directly control their influence on the overall decision.
- Whereas for costs a very low level is the optimum, for most other goals high values are preferable, which forms an orthogonal decision dimension.

These top-level objectives are furthermore divided into a wide variety of second level-goals. In defining them overlapping of different subgoals should be avoided,

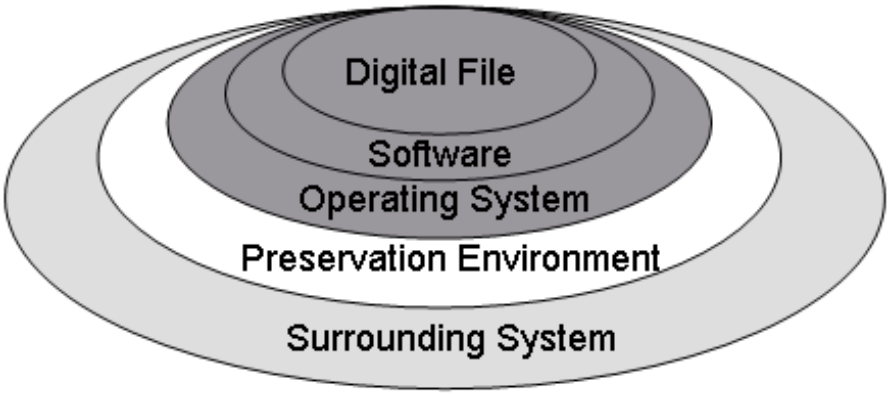

Figure 4.2: Separation between the digital object, the preservation environment, and the surrounding software

although this is not a condition for the Utility Analysis. As can be seen in an overview in Table 4.1 these three first level goals are divided into nine second level goals. I suggest the following criteria, which cover all aspects at this level, the deeper hierarchy can only be shown exemplarily.

The first aspect 'File characteristics' is sometimes mentioned shortly in literature, for example by Hans Hofmann and Jacqueline Slats who used the division into content, context, appearance, structure and behaviour [45], but has not been elaborated to a deep level of detail. Concerning the proposed factors, but also introducing a new hierarchy, I suggest to use the following differentiation of level two criteria:

- **Appearance:** The appearance is in the centre on interest for the majority of digital objects. Documents, presentations, but also websites have a strong focus on their visual impression. Under this topic all aspects of passive apprehension are covered. As soon as the user gets actively involved and influences the program calculations, or if the file itself executes any type of background calculations, this is covered in the 'Behaviour' chapter.

  Here subgoals are divided according to the species. First documents, therefore the subgoals 'Page', 'Paragraph' and 'Character' are recommended. These are again separated into fourth level criteria, such as page borders or numbering, into paragraph formation or character font style and colour. For the category 'Sound' the subgoals such as the bit rate, seem appropriate, and for moving images files the category 'Video'. A very specialized additional case would be the readability through the touch sense for blind users with Braille buttons. Due to its very special focus, this objective is not mentioned in the exemplary overview, but can without problem be added if required.

- **Structure:** Another second level goal refers to the inner structure of a file

which is not visible but which is necessary for processing it. An example is the recognition of captions in Word. If this information gets lost, the automatic creation of 'tables of content' is not possible any more. Another example is the notation with XML, where a loss of the structure makes the files unusable.

- **Behaviour:** As mentioned above in the 'Appearance' item, behaviour covers all aspects, where a file reacts on user inputs or where background logic is executed. An example is a search operation or a word count function. Another one is the execution of background logic, e.g. a MS Excel calculation. A further aspect is the context between various files, which depends on the amount of files stored at the same time - the interoperability between the stored files - and on other means, such as link databases which can additionally improve the context maintenance.

The second top-level goal 'process characteristics' is more complex to describe, because it is neither as visible as file characteristics nor well explored. Nevertheless the following four subgoals which cover all relevant aspects in this level were found. As before, the lower levels are only mentioned with examples, and as before, for specific project requirements it might be necessary to modify even second level goals.

- **Authenticity:** This aspect covers all aspects which concern the recognition, prevention or traceability of changes. It has to be considered that it is not always desirable to prevent the file from any changes.

- **Stability:** Under this topic a wide array of diverse characteristics is collected. One criterium of stability may be the independence from hard- and software producers, who might endanger the preservation project through stopping the support for one of their products. Another one is the completeness of the content during the transformation process: Is the process reliable enough to provide the same quality of output for every input and to handle possible exceptions. There are surely more third level criteria which were not mentioned here, but these are subject to real world implementations of the objective tree.

- **Scalability:** Many preservation projects do not only want to store the files which already exist at the time of the process' roll-out, but also files, which are to be created in the following years. Another aspect of scalability is an increase in the variety of stored file formats.

- **Usability:** The last part of the preservation process is its usability. Here characteristics, such as process complexity or its functionality are being described. Since easy usability is strongly linked to costs, this characteristic might be determining for many decisions.

Top level	Level 2	Level 3 (selected level 4 criteria)
File Characteristics	Appearance	Page (borders, numbering, ...)
		Paragraph (formation, ...)
		Character (font style, colour, ...)
		Sound (bit rate, ...)
		Video (frame rate, ...)
	Structure	Caption, tag description, ...
	Behaviour	Reaction on user inputs, search, ...
Process Characteristics	Authenticity	Tracebility of changes, ...
	Stability	Supplier independency, ...
	Scalability	Data increase, format range increase, ...
	Usability	Process complexity, functionality, ...
Costs	Technical	Ingest, Storage, ...
	Personnel	Installation, maintenance, ...

Table 4.1: Objective tree: Hierarchical order of goals

The third top-level goal are costs. It is usual to divide costs into technical and personell costs as second level objectives.

- **Technical costs** mean expenses for technical devices. A separation related to time when such an investment is necessary, seems beneficial. This means to use 'Ingest', 'Storage' and 'Access' as sub-predicaments.
- **Personnel costs:** The same separation is suggested for the personnel costs, which refer to the costs of development or installation and of maintenance.

The final hierarchy's depth depends on the criteria's complexity and on the user's ability of finding exactly measurable subgoals.

Users might combine this with the bottom-up approach, taking a look at the actual files in their collections, listing all relevant document characteristics to be preserved (such as page numbering, colour, links, resolution of images, presence of macros, etc.), and sort them into the previously defined top-down structure. The resulting objective tree may be rather extensive and complex for heterogeneous preservation settings, with some parts being common to many preservation initiatives, whereas others are very specific for a given collection. One example of an implemented objective tree can be found in the following subsection, where the requirements for a Word 2002 collection are being elaborated.

The first part of the table, the description of file characteristics, can be eventually used in digital libraries to standardize the attributes of a file when it is registered and described. This replaces unstable descriptions with the subjective words of the curator who works with the file, as the situation is today. Another

application is the testbed presented in the previous Chapter 4, which suggests the description with metadata, but only in addition with an open text field in order to precisely characterize the file attributes.

Example of the creation of an objective tree

Together with the journal editor of the 'Zeitschrift der Savigny-Stifung für Rechtsgeschichte (ZRG)' [19] the author developed a preservation strategy for the electronic versions of the journal. The media data are the following: The journal is published once a year, consisting of three volumes, the Germanistic section, the Canonistic Section and the Romanistic section. Together they consist of around 2.500 pages; images are included rarely, but every volume has a table of content and every article gives bibliographies. The editor's aim is to preserve the files which now exist in MS Word 2002 on her computer, in order to be able to find articles and authors in the future with an electronic search device. A second reason for considering electronic preservation is to reduce the storage space of print outs.

In order to define the objective tree, a workshop was held. The first twenty minutes were necessary to let the user understand the purpose and the content of the objective tree. Therefore in a first step the amount of the data was defined that have to be preserved, and the amount of additional files to be expected during the next years. Next the two previously mentioned targets were discussed in detail in order to widen the user's perspective. Here the discussion already got very interesting and a lot of criteria focussing on file characteristics were found. To shift the attention to other top-level criteria, they were posted on a flip chart and more criteria were collected.

The outcome was a mazy array of characteristics, not structured in any way, as can be seen in Figure 4.3 (in the figure, four top-level criteria are visible, because in the earlier version of the objective tree, 'Usability' was used as a fourth entry. One of the results of this workshop was to include 'Usability' into 'Process Characteristics').

Next the structuring of the criteria started. This was done at the same time from two sides: firstly top-down by introducing second level goals; and secondly bottom-up by assigning the criteria selected before to the different criteria groups. In the following discussion of the second level criteria, many new aspects were found and inserted into the growing decision tree, whereas others were taken out. So 'Independence from suppliers' became 'Independence from hardware' and 'Independence from Software'. After having introduced many criteria, the final tree, which can be seen in Figure 4.4, was defined. The postings on the upper left side describe the size and type of the preserved data. The unstructured postings on the lower right side were replaced by others in the hierarchy. In the upper centre, one leaf is visible with a bar on it. The colour was used to mark very important criteria, in order to facilitate the weighting in the following steps.

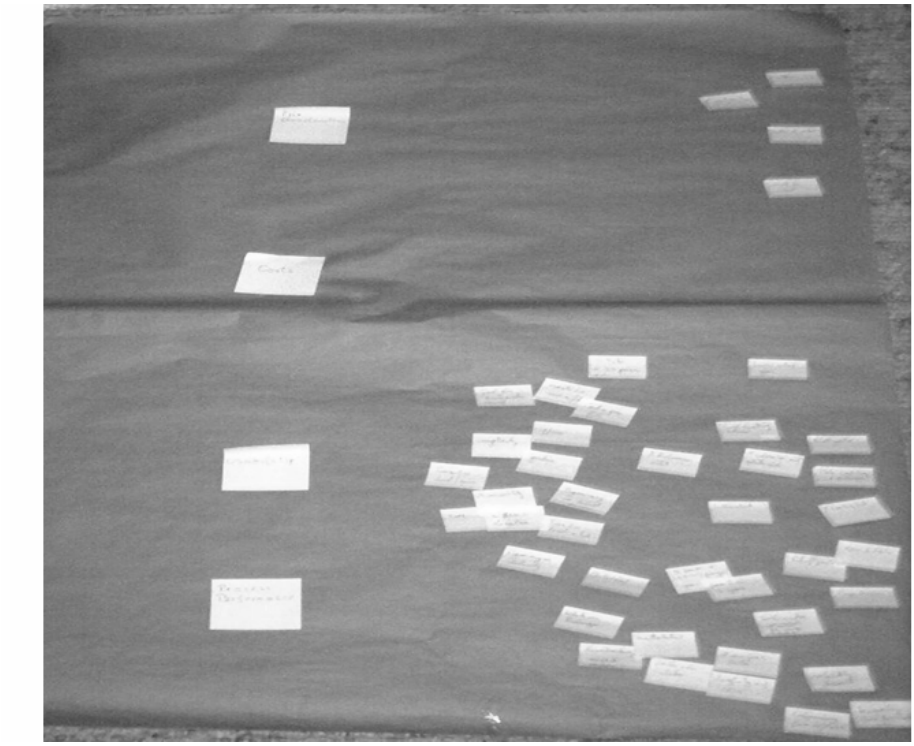

Figure 4.3: Initial non-structured collection of criteria

As homogeneity of these files is high and requirements are very narrow, some of the characteristics of the generic objective tree required no further separation into subtargets, whereas others were subdivided up to a fourth level. An example for a poorly described attribute is the structure of files, which was of no importance to the user, not using metadata descriptions and not expecting any recreation of the table of contents. On the other hand a lot of attention was paid on the availability of the file in the future, with a detailed separation concerning the portability, the hard- and software environment and worldwide access via Internet.

These criteria have been transferred into Table 4.2. The tree of the previous fourth top-level criterion is integrated in the process characteristics. The ongoing workflow of this practical example is mentioned and described in every step of the model.

4.3 Making objectives measurable

The criteria of the objective tree created in the previous step are now assigned to measurable effects using the scaling shown in Table 4.2. It does not matter, whether the targets are in the second, third, or fourth level of the tree, but if they are leafs and not nodes. The value of the nodes are defined in a later step

Top level	**Level 2**	**Level 3**	**Level 4**
File Char.	Appearance	Letters	Size
			Special characters
		Paragraph	Separation
			Picture inclusion
		Page	Footnotes
			Page numbering
			Page margins
			Page break
	Structure		
	Behaviour	Word functionality	
Process Char.	Authenticity	Change traceability	
	Stability	Completeness	Authors
			Tables of content
			Rest
		Independency from	Electricity
			Hardware
			Software
			Location(downloadable)
			Location(portable)
	Scalability	File format range	
		Size/amount of files	
	Usability	Availability	Loading time
			Saving time
		Complexity	Saving
			Maintenance
			Reopening
		Functionality	Automatisation
			Search function
			Additional remarks
Costs	Technical	Hardware	Required Space
			Electr. Energy
			Initial Assets
			Maintenance
		Software	Initial Assets
			Maintenance
			Adoption
		Personnel	Introduction
			Maintenance
	Personnel	Initial Saving	
		Maintenance	
		Reopening	

Table 4.2: Exemplary implementation of an objective tree

Figure 4.4: Final structured objective tree

by aggregating single leaf values.

Theoretically ordinal, cardinal, and proportional scales can be used for Utility Analysis. In reality usually cardinal measures such as EURO per year for hardware costs are the rule. If this is not possible, categorizing measures can be applied. Here the user chooses a value according to his impression of a criterion's fulfilment. There are no limitations on the use of scales, they can as well consist of two or of 1.000 intervals. The worst alternative choice is always the 'not acceptable' possibility. If this is chosen, the result in the objective's field changes in a way that the evaluated solution is unusable. If such a possibility is chosen, the alternative is not ranked any more in the final evaluation, but listed in the area of not sufficient solutions. Nevertheless, also negative solutions are, if it is of interest, measured in all categories to get an idea of their general potential.

In Table 4.3 some representative criteria of the objective tree created before are collected and assigned to effects. In contrast to usual evaluations, with Utility Analysis many subjective or personal decisions are required. Some of them can be answered with simple yes/no decisions, others, such as the font style is more easily evaluated with a categorizing assessment. The complete list of all criteria can be found in Table 7.1 in Appendix A.

Objective	Effect
Page numbering	correct/incorrect
Page margins	millimetre
Picture inclusion	categorizing
...	...
Software Initial Assets	€/ year
...	...

Table 4.3: Assignment of effects to objectives (excerpt, full listing see in Table 7.1 in Appendix A)

Assignment of effects to the exemplary project

The assignment of effects is also the second step in the practical example. Depending on the user's interest in technical issues, the focus is strongly oriented on measurable or categorizing evaluations. During the assignment process new criteria are added or others split into more detailed ones when several necessary measurements are found. After some practical implementations, such as this one of the law journal 'ZRG', it is possible to assign effects automatically to most objectives and to reduce the user's workload to controlling these suggestions. Because of its length the table with the assigned measures can be seen in the Appendix on Table 7.1.

4.4 Listing alternative strategies

After the definition of the objective tree and the assignment of measures for the single criteria, which helps to obtain a clearer picture of the project's perspective, the next step is to search for different preservation solutions. Alternatives have to differ significantly between each other and are to be verbally described with their names and a short overview of the preservation process. This is done for control to assure that all alternatives are understood by the project team. In addition to possible alternatives, the status-quo is defined and added, as well as the case where no planning process is made, the zero-planning case. Due to the fast technical evolution and very different user environments the listing of alternatives changes significantly in each implementation and is based on an extensive market analysis.

For the practical example of preserving the Word 2002 collection, the following four alternatives are evaluated.

1. Migration to MS Word 2003: In the example, the MS Word 2002 files are migrated to MS Word 2003.

2. Migration to the OpenOffice.org 1.0.3 (1.0.3) Writer format: The here used software is downloaded from http://www.openoffice.org. No additional automatisation steps are programmed. Because of its open specification, it is easier to decipher the code of such a file in the future with the help of a specification.

3. Migration to PDF 1,4 with the Acrobat Distiller 6.0.0: This conversion is done by printing the files with the pdf Distiller.

4. Not making any changes.

5. The zero-planning alternative is not evaluated, because no solution is used at the moment and so the results are the same as for the status-quo alternative.

Other possibilities worth considering might include conversion to level-1, level-2 Postscript files, with the possibility of migration to PDF later-on based on the PS-file, migration to pure ASCII-text, and others, as well as the separate handling of different tools for the respective steps.

4.5 Measuring the strategies' performance

In this step, the real test work has to be done. Every alternative is tested with a couple of representative files and evaluated according to the criteria of the objective tree. Some testbeds are available or under construction, where well described files in many different formats and types can be downloaded, such as in the Berkley Digital Library Project [30] or the one presented in Chapter 3. Alternatively, representative files from the collection to be preserved can be used, although care has to be taken that these are really representative with respect to the variety of document characteristics, e.g. if they include equations, embedded images of various types used in the collection, etc.

The evaluation was performed on the personal computer of the editor with a CD backup. So the hardware limits are the same for all alternatives and the costs are rather low because of no further required adoptions.

The transformations are made exactly in the way, in which they are specified in the list of alternatives in the previous section. In a first step representative files are chosen, which contain all relevant criteria, which are to be preserved. These files are then transformed and the results are evaluated and stored, as can be seen exemplary in Table 4.4 and in the full version in Table 7.2 in Appendix A. For the categorizing choice, a range from zero to five is chosen.

When the evaluation of a certain characteristic is not possible because a given file does not exhibit this characteristic (say, an animation or sound embedding), the criteria are assigned the same values for all alternatives. Because of the

Objective \ Strategy	MS Word	OpenOffice	PDF	no changes
Page numbering	yes	yes	yes	yes
Page margins	0	+3	0	0
Page break	5	N.A.	5	5
Software Initial Assets	0	0	50	0
.....	...	...	...	...

Table 4.4: Performance of the four different preservation alternatives (excerpt, full listing see in Table 7.2 in Appendix A)

equality of all possible solutions, this does not influence the final choice for this particular document.

Evaluation of the practical example

For this evaluation, some representative files of the MS Word 2002 collection where chosen and tested with the various alternative tools and processes. The files were chosen out of different working areas of the editor - one bill, one adress list, three different parts of books coming from different research areas and thus containing different special characters and also different page styles. The files were evaluated by migrating them into the various possible formats and by evaluating their appearance and behavior according to the before constructed objective tree.

In the first evaluation line of Table 4.4 all alternatives get the highest score, because all of them fulfil the requirement, they counted the pages correctly. The first differences appear at the page margins, which changed for 3 millimetres in the Open Office environment. In this alternative also the paragraph structure alters, namely in such a significant way, that the outcome is not acceptable any more.

Due to its obvious importance on the final outcome, the evaluation of the different preservation solutions for the practical example is described in more detail. The final table with all measurements can be found in Appendix A in Table 7.2.

Letter size: PDF is ranked slightly worse than the others, because of its reliance on vector graphics, which reduce the sharpness of the letters.

Special characters: It is noteworthy that also OpenOffice gets the highest ranking here, because all necessary test characters were displayed like in the original version.

Page margins: The margins varied for 3 mm in the OpenOffice environment, which directly influences the text's page break.

Page break: For the OpenOffice solution this measurement is so bad that the 'Non Acceptable' evaluation was chosen. The page breaks differed significantly

from the original, and a citation which points to the original pages is not correctly resolved any more.

Structure: PDF got only 3 as a ranking for the structure, because although the appearance is not changing the metadata information is not visible and not useable any more. MS Word got a four here, because of a couple of new tags, which are used to describe the structure, such as 'Justified'.

Word functionality: This criterion clearly prefers the no-change alternative. PDF is the only one evaluated with 1 because it offers no Word functionality. The OpenOffice solution was evaluated with 3 because it offers a similar, but not the same functionality as the original MS Word 2002, such as different kinds of viewing a file in MS Word. Also the Migration to MS Word 2003 is ranked with 4 because the functionality is changing in some ways and different to the currently used version.

Change traceability: Evaluated alternatives where no authenticity algorithms are used get only 1 as an evaluation. Only PDF is ranked with 3, because many people do not know how to change a PDF file, which makes this format somehow saver.

Software: Viewers for all alternatives exist, but PDF is ranked with four only, because of the different appearance of its files on different operating systems. The same file shown on a Linux system differs slightly according to its font and font size in comparison with Windows XP. Also the no-changes solution is independent from any new software. On the other hand, the migration to MS Word versions in the future or the reopening of the OpenOffice file depends on their specific vendors: MS Word because of the Migration steps, and OpenOffice because of its usage of a specific standard. Both are difficult to interpret by other programs.

File format range: With this criterion the scalability of the solution to other file formats are measured. It is usually not a major weight for the final outcome, but in case of doubt, it is better to have a more flexible solution. The no change alternative is measured with 5 here because all file formats can be stored and they are not touched in the future. Also PDF is ranked well with a 3 because the appearance of many files can be captured by printing with the PDF Distiller. On the other hand, the MS Word and the OpenOffice alternative are focused on text files and both of them evaluated with 3.

Storage space: For the evaluation, the storage space required for the five evaluated files was measured. It turned out, that the results varied significantly, ranging from 180 KB for the OpenOffice alternative to 260 KB, if the files are not changed. The PDF files needed 228 KB and finally MS Word 2003 208 KB.

Maintenance complexity evaluates the difficulties for preserving a solution over a longer period. The effort required for the no change alternative increases tremendously, so this solution is ranked with Not Acceptable here, as well as in the following criterion 'Reopening complexity'. The maintenance of the OpenOffice alternative is only 3 because availability of this software in five or ten years is uncertain and whether it can be found somewhere at all. On the other hand it

is probable, that Microsoft and the Microsoft Word software will still exist for a longer time and will also, to a certain degree, remain backward compatible. Maintenance covers all effort, which are required to enable the accessibility of a file, such as keeping the hard- and software running.

Automatisation: None of the alternatives can be processed automatically, so all are equally ranked with two, which because of its equality does not influence the final ranking.

Initial assets: The initial software costs for the PDF alternative of € 50 are higher than for the others because a transformation software such as the PDF Distiller is required, whereas the OpenOffice software is Freeware and MS Word 2003 is bought equally if the alternative is chosen or not, not causing any extra costs.

Personnel costs: The costs for ingesting, maintaining and reopening a file are set here according to the time needed for one conversion and a calculation of a technician cost per hour of around € f50.

The whole list with the results for the practical implementation is given in Appendix A in Table 7.2.

After the measurement of the various criteria, the result is a table with 'the number of leafs' times 'the number of alternatives' values, which are measured in different categories, such as EURO, minutes, or categorizing estimations. The next step is to transform these values into comparable numbers.

4.6 Transformation into comparable numbers

To this end, all previously obtained results are transformed to a uniform scale, which can be from zero to five, as in our example, but also in a bigger or smaller scale. It is useful to work with the same range as it is used for the categorizing evaluation of characteristics, because then the results can be directly taken as comparable numbers.

The transformation is more difficult with cardinal scales. For such values, two possible transformations are used in practice: The first is to define certain intervals which correspond to certain rankings. The second possibility is to dedicate the best result to the highest result and to scale other values linearly. In this paper the first approach of defining intervals was chosen. Table 4.5 shows the transformation function for some previously defined values, again the full table for all considered objectives is defined in Appendix A in Table 7.3. These values may differ significantly from other user's perceptions. Special care is needed when setting the 'Not Acceptable' value. Due to the discrimination of alternatives which do not fit in the intervals higher than zero, a too strict handling reduces the number of possible alternatives strongly.

Especially neither fix nor variable costs can be generally categorized because they depend directly on the collection's size.

Objective	Val. 5	Val. 4	Val. 3	Val. 2	Val. 1	N.A.
Page numbering	YES	4	3	2	NO	N.A.
Page margins [mm]	0	1	2	3	4	> 4
Page break	5	4	3	2	1	N.A.
Software Initial Assets	0	]0;20]	]20;40]	]40;80]	]80;150]	> 150
.....	...	...	...	...		

Table 4.5: Transformation of measured values to a 5 to N.A. scale (excerpt, full listing see in Table 7.3 in Appendix A)

Objective \ Strategy	MS Word	OpenOffice	PDF	no changes
Page numbering	5	5	5	5
Page margins	5	2	5	5
Page break	4	N.A.	5	5
Software Initial Assets	5	5	3	5
.....	...	...	...	...

Table 4.6: The comparable values (excerpt, full listing see in Table 7.4 in Appendix A)

In principle, the definition of the transformation functions can take place immediately after the definition of the measurement scales. Still is it recommended to define the transformation function only after the performance measurements of the various strategies have been made. This is in order to first get an overview of the scope of the values, such as e.g. the displacement of page margins in the example listed in Table 4.4. After applying the transformation functions we obtain the results listed in Table 4.6. These values form the input to the final rating. Again, the real world results of the example can be found in Appendix A in Table 7.4.

4.7 Weighing the objectives

The output of the previous step is a large table with the size of "number of alternatives" times "number of characteristics". In this step the numbers are aggregated to a single value per alternative while allowing different weighing of the various objectives. The first part is to choose the relative importance of the four top-level criteria "File characteristics", "Costs", "Usability", and "Process performance". Next every level of each branch is weighted in the same way. The next step thus is to choose the relative importance of 'Appearance', 'Structure' and 'Behaviour'. The process goes on like this until all leaves and nodes have a specific weight. An

exemplary weighing of the representative criteria can be found in Table 4.7 and the full overview of the practical example in Table 7.5. These weights depend largely on individual judgements and are to be done by the applying customer only, requiring adoption for every practical implementation. During the practical example, the definition of the individual weights worked rather fast and without complications, because the user was sure about her priorities.

Weights are defined interactively in a brain-storming session, evaluating the outcome of different decisions and their effect on the usability of the collection in the future. With some simple calculations, the final ranking can be obtained. The first part is to multiply the objective values of Table 4.6 with the objectives' weights. The outcome is called part-value. The values presented in Table 4.7 are chosen categorizingly and only reflect the requirements of our specific preservation scenario.

Finally, the weights of the single leaves can be obtained by multiplying their value times the weights of their preceeding nodes. For example, the weight of the criterion 'Numbering of chapters' is to be multiplied with the weight of 'Pages', 'Structure' and the weight of 'File characteristic'. By this way, the weights for all leafs can be calculated, summing up to 1 for each individual branch. While, again, these weights could be set immediately after defining the objective tree, it is advisable to discuss them after evaluation, thus taking the performance of the various preservation strategies into account. Weights are determined individually for every implementation of the Utility Analysis.

The total weights' aggregation of the first scenario can be found exemplary in Table 4.7 in the last column 'w.' and a complete version in Table 7.7 in Appendix A.

4.8 Ranking of alternatives

Finally we arrive at the last step, the ranking of the alternatives. First, the total weights have to be multiplied with the transformed values to get the final values for each objective, as shown in Table 7.8 in Appendix A. These values are summed up to get the final values for the single alternatives, which can be seen in Table 4.9.

A very interesting fact of this final outcome is the high value for the "no changes" alternative. This can be explained with the fact that the objective tree of the practical example considers mainly aspects of the very first phases of a preservation project. Other requirements, such as long-term stability or the long-term preservation effort, are only mentioned, but not elaborated to further detail. These are also the two categories where the "no changes" alternative turned out to be "Not Acceptable". Everywhere else in our evaluation, it has the benefit that no work for ingest is necessary and that all characteristics may be reproduced exactly.

Top level	%	**Level 2**	%	**Level 3 (selected)**	%	**w.**
File Characteristics	30					
		Appearance	70			
				Page numbering	12	0.0252
				Page margins	4	0.0084
				Page break	12	0.0084
				...		
Process perform.	40					
		...				
Costs	10					
		Technical	30			
				Software Initial Assets	5	0.0015
		...				

Table 4.7: Percentages and weights of the objectives (excerpt, full listing see in Table 7.5, in Table 7.6 and in Table 7.7 in Appendix A)

Objective	**Actual MS Word**	**Writer**	**PDF**	**no changes**
Page numbering	0.06	0.06	0.06	0.06
Page margins	0.06	0.024	0.06	0.06
Page break	0.072	N.A.	0.096	0.096
...	...	...	...	...
Software Initial Assets	0.064	0.16	0.16	0.16

Table 4.8: Part-Values (excerpt, full listing see in Table 7.8)

Before finally ranking the files, two more steps are done: The first is a sensitivity analysis, where it is tested how small changes in the weighting of the criteria influence the final outcome. Additionally it can be tested, how different aggregation methods of the partial values influence a final decision. Bröthaler and Krames suggest the calculation methods 'sum', 'product', 'ranking sum', 'Minimax' and 'Maximax' [20].

The numerical results of the Utility Analysis are then compared to side effects which are not considered in the calculation and in the objective tree. Such effects are among others good relationships with a supplier, expertise in a certain alternative, or individual assessment that one or the other solution might become the market leader within a couple of years. Nevertheless, the numerical evaluation of different alternatives provides a powerful tool to weigh their strengths and weaknesses and to make diverse solutions comparable.

Rank	Solution	Total-Value
1	MS Word Migration	4.175275
4	OpenOffice Writer	Not Acceptable (3.731875)
2	PDF	3.895975
4	No Changes	Not Acceptable (5.026475)

Table 4.9: Total-Values and final ranking of the alternatives

4.9 Summary

One major problem in the preservation research area is the choice of the right strategy for a certain data collection. The Utility Analysis is a good approach to cope with that complex situation. Because of its stringent process, while at the same time allowing subjective weighting and even evaluation of solutions which fail to fulfil knock-out criteria to a sufficient degree, it helps to reduce the complexity and increases the objectivity of the decisions taken. It allows the analysis of a range of scenarios, providing a high-level overview due to the hierarchical structure and aggregation of extensive lists of preservation requirements into higher-level objectives.

Additional to the theoretic description of the Utility Analysis' workflow, its usability is shown with the example of preserving the electronic version of the journal 'Zeitschrift für Rechtsgeschichte'.

5 Requirements for audio files

5.1 Introduction

The second practical example describes a completely different scenario than the first one. Here an objective tree is elaborated for an audio collection which is professionally stored and preserved by the Austrian Phonogrammarchiv (Österreichisches Phonogrammarchiv, Liebiggasse 5, A-1010 Vienna, Austria). Since the Phonogrammarchiv has been working in the area of digisation and digital preservation for years and since it is a member of the international research community in this area, participating in several programs, such as DELOS Network of Excellence or the 'Information for All-Program' of the UNESCO, the audio files are stored with profound knowledge and at a very professional technical level.

In this implementation only the objective tree as the core part of Utility Analysis is elaborated, because the Phonogrammarchiv is not planning any major changes in its preservation strategy, primarily to evaluate the performance of the current solution. Although Utility Analysis is only partly processed, the Phonogrammarchiv still has a considerable benefit by recognizing the great number of possible criteria.

In Section 5.2 the construction of the objective tree and the related workflow are presented. The meaning of the single leaves is described in Section 5.3.

5.2 Defining the objective tree

For constructing the objective tree, a workshop took place with representatives from the Vienna University of Technology and from the Phonogrammarchiv participating. The workshop lasted for nine hours. For future implementation workshops I suggest the following order of steps, which will be described and presented together with my experiences from this setting. In the following, the representatives of the Phonogrammarchiv are called applicants, and the representatives of the Vienna University of Technology are called externals.

- Introduction into the applicant's working area
- Introduction of the Utility Analysis
- Free brainstorming

- Collection of additional criteria with the help of the generic objective tree
- Assigning the collected criteria to the branches of the generic objective tree
- Constructing a new hierarchy

Introduction to the applicant's working area: In the beginning it is necessary for the externals to get some basic knowledge about the problems with which the applicant's institution is dealing. Only then they can understand the concerns and ideas of the preserving institution and so simplify the expected discussion. In our case we got to know an ingest stations, where the files are prepared and connected with metadata, and we had a longer discussion over the state-of-the-art in audio (and video, which is the second - here not treated - working area of the Phonogrammarchiv) preservation.

Introduction of Utility Analysis: The same has to be done the other way around: the applicants were introduced to Utility Analysis, to the process and to the benefits which can be expected from applying it. The applicants should identify themselves with these aims and see a benefit in participating.

Free brainstorming: Real work starts with giving all participants some time to understand the introduction and to develop slowly a feeling for the kind of information required for the process. Participants introduce criteria which they think might be relevant.

Discussion of identified criteria: These criteria are then discussed in detail in order to get a common understanding. Usually, the output of the first round is miles away from what will be finally used for the objective tree, so such criteria give a basis for further thought, but will usually not find themselves in the final objective tree.

Collection of additional criteria with the help of the generic objective tree: After a while, which is usually between half an hour and an hour, all criteria which initially come into the applicants' and the externals' mind are mentioned. Then the collection process starts to slow down. At this time the generic objective tree as presented in Chapter 4, is introduced to provide new inputs to the discussion. In the workshop at the Phonogrammarchiv we discussed every branch of this generic objective tree in detail and found many new criteria. The process lasted approximately for another 2 1/2 hours and the result is a big (chaotic) collection of criteria, as can be seen in Figure 5.1.

Assigning the collected criteria to the tree's branches: The next step is to bring some order into this unordered mass. Therefore the criteria are first discussed in more detail and then assigned to the second level criteria of the objective tree, while duplicate or overlapping criteria can already be corrected. The criteria are not being assigned to the second level criteria right during the brainstorming session for fear that it might limit the freedom of ideas. In the here described case most criteria are assigned to 'Stability' and 'Usability'. A second interesting fact is the strong connection between 'Usability' and 'Personnel Costs',

Figure 5.1: The result of the brainstorming process

as many criteria can be assigned to either of them. We solved the difficulty by assigning all criteria measurable with money to the costs criteria, and all others to the heterogenic 'Usability' criteria.

Constructing a new hierarchy: The last step is to create a deeper hierarchy within second level criteria. Here all subcriteria assigned to one second-level criterion have to be reconsidered. It should be tried to find a certain logic and connection among them. Thereby duplicates might be found, others might be formulated in another way. Finally the tree resulted, which can be seen in Figure 5.2 on a photograph after its creation and in Table 5.1, Table 5.2, Table 5.3 and Table 5.4 in a more readable form.

5.3 Description of the leaves

After the creation of the objective tree, it is useful to additionally describe every leaf in detail in order to avoid misunderstandings and to clearly specify what is measured with which attribute. During the creation process of the tree, it frequently happened that the participants understood different attributes by reading the same criteria. Therefore each criteria's objective and a possible way to measure it are presented here:

Audio resolution: Describes the band-width, which is measured in bit per sample.

Sample rate: Describes how often per second the data source is touched. This also influences through the Shannon theorem the highest frequency that can be reproduced.

Compression rate: Decribes the degree to which a file is compressed and if it is saved lossy or lossless.

Loss through watermarking: An additional source of malfunctions may be a watermarking, which is included directly into the sound. This may reduce the quality of the file. The existence of water marking in a positive sense is described with another criterion.

Sound replay: Concerns the sound's technical possibilities, such as Stereo, and is probably measured with a subjective scale or with specific quality grades.

Replay velocity: With this figure it should be determined, if a formatted unit is still played in its original speed or if it got a little too fast or too slow.

Marker: This and the following objectives describe different metadata attributes which are not necessarily part of the file's metadata description. The markers are time points within an audio file, where the user directly jumps to. Such markers are usually set at scene changes.

Source: Measures, whether a description of the file, which was transformed into the stored file, is available.

History: The history of the file, starting at the time of creation and taking into account any change during its lifetime.

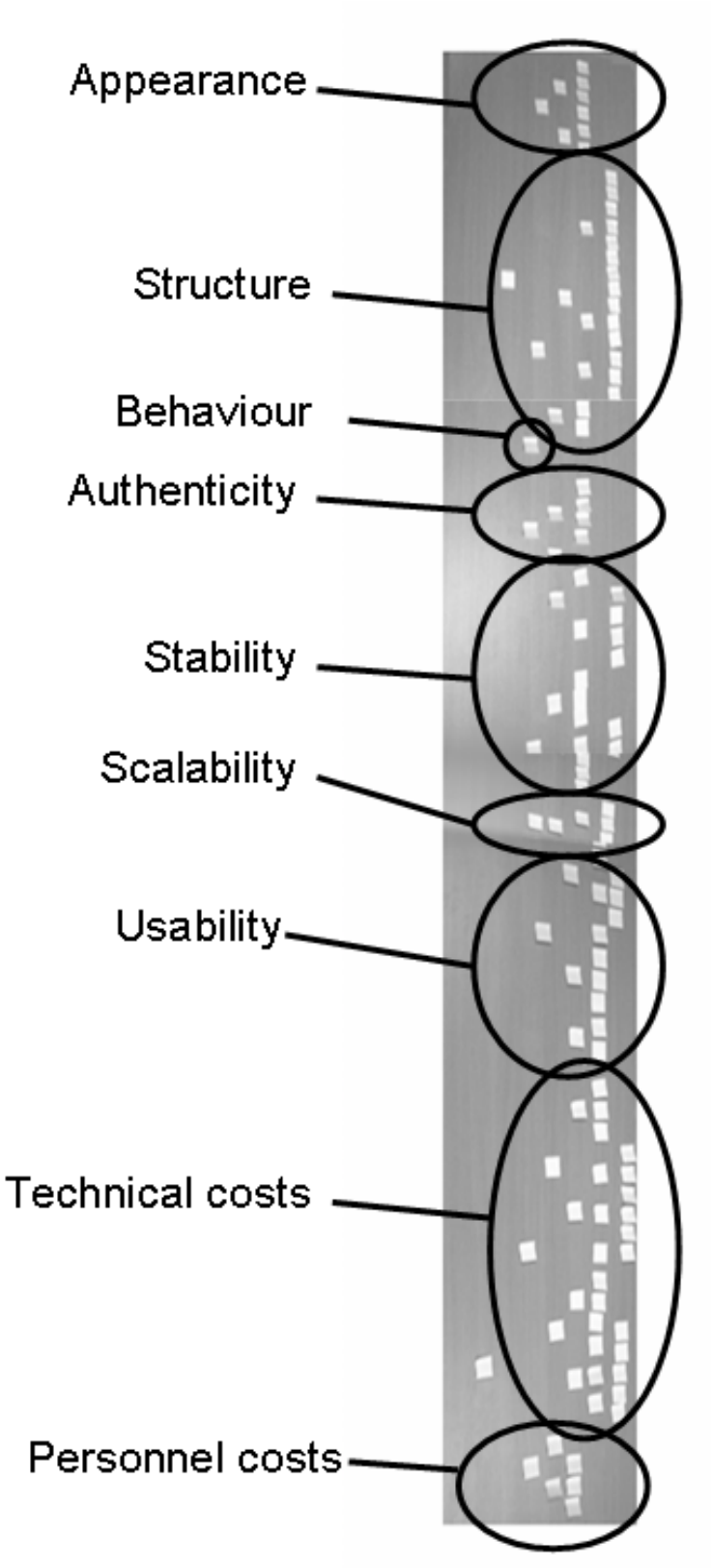

Figure 5.2: Final objective tree for the Phonogrammarchiv

Time Code Information is an exact measure of the time within a file. With the help of it, every scene can be attached to an exact time, which might be important for audio-video synchronisation.

Technical equipment: Measures, whether the technical equipment which was used to migrate the file, is described.

Title: Indicators, whether the title of the digital object is preserved.

Interpret: Indicators, whether the actors and players of the audio file are described.

Copyright: Indicators, whether copyright questions are described in the metadata collection.

Link to external metadata-database: This and the following two criteria describe how metadata are assigned to the original file. This first objective defines, whether the metadata are stored in a separate database.

Metadata included in the file: Here it is described, whether the metadata are stored directly in the audio object, for example in the header of the file, or as tags within the file.

Metadata encapsulated with the file: Another possibility is to encapsulate the files together, but storing the information in different files.

Import of metadata from original file: This line states how metadata are obtained. The first possibility is the direct import from the original file.

Automatic metadata extraction: A second possibility is to extract metadata automatically. It might seem that some objectives exclude one another, but they may always be used parallel to each other. For example, some metadata can be imported from the original file, while others are automatically extracted.

Manual metadata insertion: Additional to the two previous mentioned ways of metadata generation, these data can also be inserted manually.

Storage size: Here the average amount of used storage per audio unit is measured, for example in MB/sec.

Uncompressed storage: This simple yes/no criterion describes whether the audio files are stored in a compressed or uncompressed way.

Watermarking: Describes, whether the signs, which describe the origin of a file, are preserved or not.

Storage safety: Here the physical storage of the objects is considered in short, taking fire save doors, locked containers and similar devices into account.

Registration of changes should reliably register all changes which occur to the object.

Signatures: The use of signatures to detect changes.

File format verification: The use of tools to verify whether the description of a file corresponds to the real content. Technically such a verification is only possible for a few file formats.

Multiple storage media indicates, if different forms of storage are used to gain an improved security through redundancy.

Duration of guarantee refers to the duration, where the hardware vendor guarantees the functionality of his devices.

Estimated duration gives the tested and reported lifetime of the hardware storage devices.

Industry support: Refers to the duration in which a specific product is supported by its producing company.

Backwards compatibility: Describes for how long a file or tape can be opened with future soft- and hardware generations. Here the information given by the producer is used.

Hardware independency: The extend to which the storage is independent of a specific hardware environment.

External control and access: The access to the storage devices via Internet in order to be able to control the processes without being directly there.

Portability: The possibility to move the storage media in case of an emergency, but also in case of a relocation.

Tape support indicates, if the used kind of tape is supported by the software.

Storage structure: Considers the specific storage structure, which the software applies. Here the length of the location identifier can vary.

Migration complexity specifies the amount and difficulty of work for migrating a file.

Open format shows public availability of a file documentation and dependency of the format and the software, which works with the format.

Spread of the format: Decribes the use of the format, e. g. whether its marketshare is 80 or 90 percent, such as WAVE or on the other extreme a self-programmed format, which is not used anywhere else in the world.

Robustness (Error Tolerance): Describes, how a file reacts on minimal errors, such as the change of one bit. In some cases it might not be useable any more, in others the error is neglectable.

Automatic error recognition: Here it is measured whether errors may be automatically detected or not.

Workload of computer: This objective describes up to which limit the computer is capable of processing digital objects in a reasonable time.

Technical limit of scalability: Different storage solutions allow different amounts of files to be stored. This should be considered with this criterion.

Commercial limit of scalability: On the other hand there is also a commercial limit up to which an upgrade of the chosen storage solution makes sense and from that on another storage solution would be more useful.

Format range scalability: The possibility to store other file formats as well or the limitation to a specific format.

Metadata ingest speed: The time which is used for storing the additional information, such as metadata.

Need for technical assistance: Whether the assistance of a technician is required for the ingest. Some criteria, like this one, appear more than one time

in different branches, but with the same meaning. In this case they will only be described here once.

Time to understand the system: The average time, which is needed by a new user to understand the storage system.

Speed of ingest: The speed needed for the ingest of the file itself, possibilities would be in real-time or in a faster mode.

Format profiles: The possibility of creating ingest profiles for different file formats to avoid to re-enter all specific values every time the format is changing.

Speed of datatransfer to archive: Whether the files are transferred online or manual to the archive and how much time it costs.

Amount of automatisation: The degree to which the maintenance process can be automated, for example with format recognition systems and automatic storage media copying.

Frequency of migration cycles: Although this criterion cannot be determined exactly, it can be estimated according to the behaviour of the solution in the past. It estimates, how often (in years) a migration cycle of the whole storage will be necessary.

Reading speed of tape: The time needed to read the audio file itself.

Reading speed of index: The time to read a file's index.

Opening speed online: The time needed to read a file which is stored in an online deposit, such as a server.

Opening speed nearline: The time needed to read a file which is stored nearline, such as in a band container, which can be opened automatically.

Opening speed offline: The time needed to read a file which is stored offline, e.g. in a separate area, where files can only be opened with human help.

Preparation of often mentioned files: Whether often mentioned files are stored separately to enable a faster access.

Multiple data access: The possibility to read several data from the storage at the same time.

Partial decompression possible: If it is possible to extract only a part of an electronic file in order to spare time when reading a part of it.

These objectives, which are measured by costs, are straightforward and do not require further description. In the first technical part, all costs for hard- and software are covered, whereas in the second part all personell costs are included. Here only audio-specific attributes are described:

HSM Software: This software organizes the automatic back-up of unused files and also automatically reopens them from tape, when they are requested.

Backup Software: The less powerful alternative is the backup software, which only transfers data to tape after an explicit user order.

Tape cartidge: Tapes are often used for storing large amounts of audio data.

5.4 Summary

In this chapter only the first step of Utility Analysis is elaborated. Thereby many criteria, which could influence the choice for a preservation solution are defined and sorted into an objective tree. Thereby not only those, which directly concern the preservation solution are enlisted, but also criteria, which are only slightly affected by one or another preservation approach.

By creating the objective tree, the major part of the Utility Analysis is done, which usually requires most time of the evaluation process. Besides being the basis for the analysis, creating an objective tree is also a powerful tool to identify and structure many different aspects of a digital collection.

Top level	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**
File Char.	Appearance	Quality	Audio resolution		
			Sample rate		
			Compression rate		
			Water marking		
		Functionality	Sound Replay (e.g. Stereo)		
			Replay Velocity		
	Structure	Metadata	Content	Marker	
				Source	
				History	
				Time Code Information	
				Technical Equipment	
				Title	
				Interpreter	
				Copyright	
			Kind of connection	Link to external metadata-database	
				Metadata included in the file	
				Metadata encapsulated with the file	
			Kind of generation	Import of metadata from the original file	
				Automatic metadata extraction	
				Manual metadata insertion	
		Requ. Storage	Storage size		
			Uncompressed storage		
	Behaviour				

Table 5.1: Objective tree for the Phonogrammarchiv (1/4)

Top level	Level 2	Level 3	Level 4	Level 5	Level 6
Process Char.	Authenticity	Traceability of changes	Watermarking		
			Storage safety		
			Registration of changes		
			Signatures		
		File format verification			
	Stability	Storage	Multiple storage media		
			Durability	Duration of guarantee	
				Estimated duration	
				Industry support	
				Backwards compatibility	
			Hardware independence		
			External control and access		
			Portability		
			Software	Tape support	
				Storage structure	
		Format	Migration complexity		
			Open format		
			Spread of the format		
			Robustness (Error Tolerance)		
		Autom. error recognition			
	Scalability	Storage capacity	Workload of computer		
			Limit of scalability		
				Technical limit	
				Commercial limit	
		Format range scalability			

Table 5.2: Objective tree for the Phonogrammarchiv (2/4)

Top level	Level 2	Level 3	Level 4	Level 5	Level 6
	Usability	Ingest	Complexity	Metadata ingest speed	
				Need for technical assistance	
				Time to understand the system	
			Speed of ingest		
			Format profiles		
			Speed of datatransfer to archive		
		Maintenance	Complexity	Time to understand the system	
				Need for technical assistance	
				Amount of automatisation	
			Frequency of migration cycles		
		Reopening	Time to user	Reading speed of tape	
				Reading speed of index	
				Opening speed	Online
					Nearline
					Offline
				Preparation of often mentioned files	
			Complexity	Automatisation	
				Toolsupport	
				Technical assistance	
			Technical	Multiple data access	
				Partial decompression possible	

Table 5.3: Objective tree for the Phonogrammarchiv (3/4)

Top level	Level 2	Level 3	Level 4	Level 5	Level 6
Cost	Technical	Initial	Software-Costs		
				HSM Software	
				Back-up Software	
				Standard software	
			Hardware	Ingest	
					Device for write/read a tape
					Station for ingest
				Storage	Tape cartridge
					Storage sub systems
					Server
				Reopening	Userstation
		Running	Software-Costs	HSM Software	
				Standard Software	
			Hardware	Ingest	Device for write/read a tape
					Station for ingest
				Storage	Tape cartridge
					Storage subsystems
					Server
				Reopening	Userstation
	Personnel	Enrolment			
		Running costs	Ingest		
			Maintenance		
			Reopen		

Table 5.4: Objective tree for the Phonogrammarchiv (4/4)

6 Conclusion

During the last couple of years, long-term preservation of digital files has become increasingly important. Institutions are recognizing the risk of not being able to store their digital files over longer periods, and start research in this area.

The thesis consists of three parts.

The first part describes the state of the art in the preservation research field. It is tried to cover all major preservation initiatives, realistic and futuristic ones, long-term and short ranging strategies. Beside well known ideas, namely Migration, Emulation and Computer Museums, also some less recognized proposals, such as the Digital Tablet, are being presented. An interesting result is the strong interdependency between Migration and Standardisation on the one hand, and Encapsulation and Emulation on the other hand. A second interesting outcome is the separation of Migration into the four substrategies 'Migrating to other formats', 'Migration on Demand', 'Migration within the same format' and 'Software Migration'.

The second chapter describes a testbed which was created as part of this master thesis. It provides a framework for storing and describing a wide array of different files and file formats which can be used to test preservation solutions.

The third part, finally, is the most significant contribution of this thesis. It describes Utility Analysis, a tool which can be used to support and justify decisions. This method, which was developed to evaluate infrastructure projects, can be applied with some small modifications for evaluating different preservation solutions. The process starts with defining the objectives of a preservation project, whereby an objective tree is created. This tree is one of the core elements of the analysis. As a next step, alternative preservation solutions are enlisted and rated according to the outcomes measured for the single objectives. Then these objectives are weighed by the user, thus allowing a final ranking.

In order to demonstrate the analysis' usability it is applied in practise on the law journal 'Zeitschrift für Rechtsgeschichte' and on the audio collection of the Austrian Phonogrammarchiv. In the first case, the whole process was implemented, whereas for the second one only the objective tree, which is the core issue of the process, was developed.

7 Appendix A

Criterion	**Effect**	**Description**
Letter Size	point	Height and width of letters
Special characters	categorizing	Appearance and correctness
Paragraph Separation	categorizing	The paragraph breaks
Picture inclusion	categorizing	Right position and quality
Footnotes	categorizing	Existence, position and appearance
Page numbering	yes/no	Existence, correctness
Page borders	mm	Space between page and side border
Page break	categorizing	Text break at the same part of text
Structure	categorizing	Metadata info on parts of the text
Word functionality	categorizing	Enter, change, delete text, ...
Change traceability	categorizing	Protocols, signatures
Authors	categorizing	List of authors at beginning of book
Tables of content	categorizing	Link to chapter starts, titles, authors
Rest	categorizing	Loss through preservation
Electricity	yes/no	Required to reopen the files
Hardware	categorizing	Dependence on certain hardware
Software	categorizing	Dependence on certain software
Location (downloadable)	yes/no	Worldwide availability
Location (portable)	categorizing	Portable solution
File format range	categorizing	No. of additional usable file formats
Storage space	MB total	Available storage space
Loading time	seconds/file	Average time to open a file
Saving time	seconds/file	Average time to ingest a file
Saving complexity	categorizing	Skills required for ingest
Maintenance complexity	categorizing	Skills required for maintenance
Reopening complexity	categorizing	Skills required for reopening
Automatisation	categorizing	Possibility of automatisation
Search function	categorizing	Functionality, availability
Additional remarks	categorizing	Possibility to add remarks
Required Space	€/year	Costs for used square meters
Electr. Energy	€/year	Approx. for the preservation solution
Initial Assets	€	Hardware initially required
Maintenance	€/year	Hardware required over time
Initial Assets	€	Software initially required
Maintenance	€/year	Software required over time
Adaption	€	Effort to adapt the software
Introduction	€	Work for HW / SW introduction
Maintenance	€/year	Work for HW / SW maintenance
Initial Saving	€/file	Work for file ingest
Maintenance	€/file	Work for maintenance
Reopening	€/file	Work for file reopening

Table 7.1: Assignment of effects to the exemplary objective tree

Objective \ Strategy	**MS Word**	**OpenOffice**	**PDF**	**No Changes**
Letter Size	5	5	4	5
Special characters	5	5	5	5
Paragraph Separation	5	5	5	5
Picture inclusion	5	5	5	5
Footnotes	5	5	5	5
Page numbering	5	5	5	5
Page margins	0	+3 mm	0	0
Page break	5	**N.A.**	5	5
Structure	4	5	3	5
Word functionality	4	3	1	5
Change traceability	1	1	2	1
Authors	5	5	5	5
Tables of content	5	5	5	5
Rest	5	5	5	5
Electricity	no	no	no	no
Hardware	yes	yes	yes	yes
Software	1	2	4	5
Location(downloadable)	yes	yes	yes	yes
Location(portable)	yes	yes	yes	yes
File format range	3	3	3	5
Storage space	208 KB	180 KB	228 KB	260 KB
Loading time	3 sec	8 sec	3 sec	3 sec
Saving time	5 sec	7 sec	8 sec	0 sec
Saving complexity	4	4	3	5
Maintenance complexity	5	3	5	**N.A.**
Reopening complexity	4	3	5	**N.A.**
Automatisation	2	2	2	2
Search function	5	5	5	5
Additional remarks	5	5	3	5
Required Space	0	0	0	0
Electr. Energy	0	0	0	0
Initial Assets	0	0	0	0
Maintenance	200	200	200	200
Initial Assets	0	0	50	0
Maintenance	25	0	25	0
Adaption	0	0	0	0
Introduction	0	0	0	0
Maintenance	15	15	15	0
Initial Saving	1	3	3	0
Maintenance	1	1	1	0
Reopening	3	3	3	0

Table 7.2: Evaluation of the different preservation solutions

Objective	**Val.5**	**Val.4**	**Val.3**	**Val.2**	**Val.1**	**N.A.**
Letter Size	5	4	3	2	1	N.A.
Special characters	5	4	3	2	1	N.A.
Paragraph Separation	5	4	3	2	1	N.A.
Picture inclusion	5	4	3	2	1	N.A.
Footnotes	5	4	3	2	1	N.A.
Page numbering	YES	4	3	2	NO	N.A.
Page margins [mm]	0	1	2	3	4	> 4
Page break	5	4	3	2	1	N.A.
Structure	5	4	3	2	1	N.A.
Word functionality	5	4	3	2	1	N.A.
Change traceability	5	4	3	2	1	N.A.
Authors	5	4	3	2	1	N.A.
Tables of content	5	4	3	2	1	N.A.
Rest	5	4	3	2	1	N.A.
Electricity	yes	4	3	2	no	N.A.
Hardware	yes	4	3	2	no	N.A.
Software	yes	4	3	2	no	N.A.
Location(downloadable)	yes	4	3	2	no	N.A.
Location(portable)	yes	4	3	2	no	N.A.
File format range	5	4	3	2	1	N.A.
Storage space [KB]	< 100	100-170	170-270	270-370	370-500	> 500
Loading time[sec]	0-4	5-7	7-9	10-13	14-20	>20
Saving time [sec]	0-4	5-7	7-9	10-13	14-20	>20
Saving complexity	5	4	3	2	1	N.A.
Maintenance complexity	5	4	3	2	1	N.A.
Reopening complexity	5	4	3	2	1	N.A.
Automatisation	5	4	3	2	1	N.A.
Search function	5	4	3	2	1	N.A.
Additional remarks	5	4	3	2	1	N.A.
Required Space [€]	0	0-20	20-40	40-80	80-150	> 150
Electr. Energy [€]	0	0-20	20-40	40-80	80-150	> 150
Initial Assets [€]	0	0-20	20-40	40-80	80-150	> 150
Maintenance [€]	0	0-50	50-100	100-150	150-250	> 250
Initial Assets [€]	0	0-20	20-40	40-80	80-150	> 150
Maintenance [€]	0	0-20	20-40	40-80	80-150	> 150
Adaption [€]	0	0-20	20-40	40-80	80-150	> 150
Introduction [€]	0	0-20	20-40	40-80	80-150	> 150
Maintenance [€]	0	0-20	20-40	40-80	80-150	> 150
Initial Saving [€]	0	0-5	6-10	11-15	15-25	> 25
Maintenance [€]	0	0-2	2-3	4-5	6-10	> 10
Reopening [€]	0	0-5	6-10	11-15	15-25	> 25

Table 7.3: Transformation of measured values to a 5 to 0 (N.A.) scale

Objective \ Strategy	**MS Word**	**OpenOffice**	**PDF**	**No Changes**
Letter Size	5	5	4	5
Special characters	5	5	5	5
Paragraph Separation	5	5	5	5
Picture inclusion	5	5	5	5
Footnotes	5	5	5	5
Page numbering	5	5	5	5
Page margins	5	2	5	5
Page break	4	**N.A.**	5	5
Structure	4	5	3	5
Word functionality	4	3	1	5
Change traceability	1	1	2	1
Authors	5	5	5	5
Tables of content	5	5	5	5
Rest	5	5	5	5
Electricity	1	1	1	1
Hardware	5	5	5	5
Software	1	2	4	5
Location(downloadable)	5	5	5	5
Location(portable)	5	5	5	5
File format range	3	3	3	5
Storage space	3	3	3	3
Loading time	5	3	5	5
Saving time	4	4	3	5
Saving complexity	4	4	3	5
Maintenance complexity	5	3	5	**N.A.**
Reopening complexity	4	3	5	**N.A.**
Automatisation	2	2	2	2
Search function	5	5	5	5
Additional remarks	5	5	3	5
Required Space	5	5	5	5
Electr. Energy	5	5	5	5
Initial Assets	5	5	5	5
Maintenance	1	1	1	1
Initial Assets	5	5	3	5
Maintenance	3	5	3	5
Adaption	5	5	5	5
Introduction	5	5	5	5
Maintenance	4	4	4	5
Inital Saving	4	4	4	5
Maintenance	4	4	4	5
Reopening	4	4	4	5

Table 7.4: Comparable and transformed values

Objective	Level 2	Level 3	Level 4	Weight
File Char.				0.3
	Appearance			0.7
		Letters		0.3
			Size	0.5
			Special characters	0.5
		Paragraph		0.3
			Separation	0.5
			Picture inclusion	0.5
		Page		0.4
			Footnotes	0.3
			Page numbering	0.3
			Page margins	0.1
			Page break	0.3
	Structure			0.0
	Behaviour			0.3
		Word functionality		1
Process Char.				0.6
	Authenticity			0.1
		Change traceability		1
	Stability			0.4
		Completness		0.8
			Authors	0.3
			Tables of content	0.3
			Rest	0.4
		Independency from		0.2
			Electricity	0.1
			Hardware	0.2
			Software	0.3
			Location(download)	0.2
			Location(portable)	0.2
	Scalability			0.1
		File format range		0.3
		Storage space		0.7
	Usability			0.4
		Availability		0.3
			Loading time	0.7
			Saving time	0.3
		Complexity		0.4
			Saving	0.6
			Maintenance	0.2
			Reopening	0.2
		Functionality		0.3
			Automatisation	0.3
			Search function	0.4
			Additional remarks	0.3

Table 7.5: Weighting of the single objectives(1/2)

Objecitve	**Level 2**	**Level 3**	**Level 4**	**Weight**
Costs				0.1
	Technical			0.3
		Hardware		0.25
			Required Space	0.2
			Electr. Energy	0.05
			Initial Assets	0.3
			Maintenance	0.45
		Software		0.25
			Initial Assets	0.2
			Maintenance	0.6
			Adaption	0.2
		Personnel		0.5
			Introduction	0.4
			Maintenance	0.6
	Personnel			0.7
			Initial Saving	0.4
			Maintenance	0.3
			Reopening	0.3

Table 7.6: Weighting of the single objectives(2/2)

Objectives	**Total weights**
Letter Size	0.0315
Special characters	0.0315
Paragraph Separation	0.0315
Picture inclusion	0.0315
Footnotes	0.0252
Page numbering	0.0252
Page margins	0.0084
Page break	0.0252
Structure	0.0
Word functionality	0.09
Change traceability	0.06
Authors	0.0576
Tables of content	0.0576
Rest	0.0768
Electricity	0.0048
Hardware	0.0096
Software	0.0144
Location(downloadable)	0.0096
Location(portable)	0.0096
File format range	0.018
Size/amount of files	0.042
Loading time	0.0504
Saving time	0.0216
Saving complexity	0.0576
Maintenance complexity	0.0192
Reopening complexity	0.0192
Automatisation	0.0216
Search function	0.0288
Additional remarks	0.0216
Required Space	0.0015
Electr. Energy	0.000375
Initial Assets	0.00225
Maintenance	0.003375
Initial Assets	0.0015
Maintenance	0.0045
Adaption	0.0015
Introduction	0.006
Maintenance	0.009
Initial Saving	0.028
Maintenance	0.021
Reopening	0.021

Table 7.7: Final total weights of the single leaves

Objective \ Strategy	**MS Word**	**OpenOffice**	**PDF**	**No Changes**
Letter Size	0.1575	0.1575	0.126	0.1575
Special characters	0.1575	0.1575	0.1575	0.1575
Paragraph Separation	0.1575	0.1575	0.1575	0.1575
Picture inclusion	0.1575	0.1575	0.1575	0.1575
Footnotes	0.126	0.126	0.126	0.126
Page numbering	0.126	0.126	0.126	0.126
Page margins	0.0336	0.0168	0.0336	0.0336
Page break	0.1008	**N.A.**	0.126	0.126
Structure	0.0	0.0	0.0	0.0
Word functionality	0.36	0.27	0.09	0.45
Change traceability	0.06	0.06	0.12	0.06
Authors	0.288	0.288	0.288	0.288
Tables of content	0.288	0.288	0.288	0.288
Rest	0.384	0.384	0.384	0.384
Electricity	0.0048	0.0048	0.0048	0.0048
Hardware	0.048	0.048	0.048	0.048
Software	0.0144	0.0288	0.0576	0.72
Location(downloadable)	0.048	0.048	0.048	0.048
Location(portable)	0.048	0.048	0.048	0.048
File format range	0.054	0.054	0.054	0.09
Size/amount of files	0.126	0.126	0.126	0.126
Loading time	0.252	0.1512	0.252	0.252
Saving time	0.0864	0.0864	0.0648	0.108
Saving complexity	0.2304	0.2304	0.1728	0.288
Maintenance complexity	0.096	0.0576	0.096	**N.A.**
Reopening complexity	0.0768	0.0576	0.096	**N.A.**
Automatisation	0.0432	0.0432	0.0432	0.0432
Search function	0.144	0.144	0.144	0.144
Additional remarks	0.108	0.108	0.0648	0.108
Required Space	0.0075	0.0075	0.0075	0.0075
Electr. Energy	0.001875	0.001875	0.001875	0.001875
Initial Assets	0.01125	0.01125	0.01125	0.01125
Maintenance	0.00375	0.00375	0.00375	0.00375
Initial Assets	0.0075	0.0075	0.0045	0.0075
Maintenance	0.0135	0.0225	0.0135	0.0225
Adaption	0.0075	0.0075	0.0075	0.0075
Introduction	0.03	0.03	0.03	0.03
Maintenance	0.036	0.036	0.036	0.045
Initial Saving	0.112	0.112	0.112	0.14
Maintenance	0.084	0.084	0.084	0.105
Reopening	0.084	0.084	0.084	0.105

Table 7.8: Final partial values of the single objectives, obtained by multiplying the final weights with the transformed values

Bibliography

[1] ADOBE Systems Inc. *PDF Reference*, fourth ed., 2001. URL http://partn-ers.adobe.com, Date of download: 10th of March 2004.

[2] Archaeology Data Service, University of York. *Guidelines for Depositors*, version 1.1 ed., November 2003. URL http://ads.ahds.ac.uk, Date of download: 15th of November 2003.

[3] Bechmann, A. Nutzwertanalyse, *B*ewertungstheorie und *P*lanung. *Beiträge zur Wirtschaftspolitik Volume 29* (1978).

[4] Borghoff, U. M., Rödig, P., Scheffczyk, J., and Schmitz, L. *Langzeit-archivierung, Methoden zur Erhaltung digitaler Dokumente*, 1 ed. dpunkt.verlag GmbH, 2003.

[5] Burnard, L., and Sperberg-McQueen, C. M. Text Encoding Initiative, Encoding for Interchange: An Introduction to the TEI, 2002. URL http://www.tei-c.org/Lite, Date of download: 3rd of September 2004.

[6] Curl Exemplars in Digital Archives, 2004. URL http://www.leeds.ac.uk/cedars.

[7] Consultative Committee for Space Data System. *Reference Model for an Open Archival Information System (OAIS)*, first ed., May 1999. URL http://www.ccds.org, Date of download: 11th of December 2003.

[8] Dublin Core Metadata Initiative. *Dublin Core Metadata Element Set*, 1.1 ed., December 2003. URL http://dublincore.org/documents/2003/02/04/dces, Date of download: 12th of December 2003.

[9] Duval, E., Hodgins, W., Sutton, S., and Weibel, S. L. Metadata principles and practicalities. *D-Lib Magazine 8 (4)* (April 2002). URL http://www.dlib.org, Date of download: 13th of December 2003.

[10] ERPANET. *erpastudies pharmaceuticals*, 2003. URL http://www.erpanet.org, Date of download: 8th of January 2004.

[11] ERPANET. *erpastudies telecommunications*, 2003. URL http://www.erpanet.org, Date of download: 8th of January 2004.

[12] G5 Processor Architecture, 2004. URL http://www.apple.com/g5processor/archi-tecture.html, Date of download: 8th of September 2004.

[13] Granger, S. Emulation as a digital preservation strategy. *D-Lib Magazine 6 (10)* (October 2000). URL http://www.dlib.org, Date of download: 14th of December 2003.

[14] GRAU Data Storage AG. *Infinistore Tape Library - Model XL*, 2003. URL http://www.graudatastorage.de, Date of download: 31st of August 2004.

[15] Guide to Archiving, November 2003. URL http://www.diffuse.org, Date of download: 5th of November 2003.

[16] Hanusch, H., Biene, P., and Schlumberger, M. *Nutzen-Kosten-Analyse*. Verlag Franz Vahlen, 1987.

[17] Hedstrom, M., and Lampe, C. Emulation vs. migration. do users care? *RLG DigiNews 5*, 6 (December 2001). URL http://www.rlg.org/preserv/diginews, Date of download: 05th of February 2004.

[18] International Standard Organization. *Document Management - Electronic document file format for long-term preservation - Part 1: Use of PDF(PDF/A)*, October 2003. ISO TC 171/SC 2 N.

[19] Knütel, R., and Thür, G., Eds. *Zeitschrift der Savigny Stiftung für Rechtsgeschichte (ZRG), Romanistische Abteilung*, vol. 120. Böhlau Verlag Wien, Köln, Weimar, 2003.

[20] Krames, G., and Bröthaler, J. *NWA-Applet - Nutzwertabalyse im Internet*. PhD thesis, Institute for Public Finance and Infrastructure Policy, Technical University of Vienna, 1999. URL http://www.ifip.tuwien.ac.at/forschung.htm, Date of download: 04th of April 2004.

[21] Kranch, D. A. Beyond migration: Preserving electronic documents with digital tablets. *Information Technologies Libraries 17*, 3 (1998).

[22] Kranch, D. A. Preserving electronic documents. In *Proceedings of the 3rd ACM International Conference on Digital Libraries, June 23-26, 1998, Pittsburgh, PA, USA* (1998), ACM, pp. 295–296. URL http://doi.acm.org/10.1145/276675.276740.

[23] Lawrence, G. W., Kehoe, W. R., Rieger, O. Y., Walters, W. H., and Kenney, A. R. *Risk Management of Digital Information: A File Format Investigation*. Council on Library and Information Resources, Washington D.C., 2000.

[24] Lorie, R. The uvc: a method for preserving digital documents - proof of concept. *IBM/KB Long-Term Preservation Study Report Series 4* (December 2002).

[25] Lorie, R. A. Long-term archiving of digital information. Tech. rep., Almaden Research Center, San Jose, 2000. IBM Research Report RJ 10185 (95059).

[26] Myfileformats. Website, 2004. URL http://www.MyFileFormats.com, Date of download: 1rst of September 2004.

[27] National Information Standards Organization. *Permanence of Paper for Publications and Documents in Libraries and Archives*, 1992. An American National Standard, Approved October 26, 1992, the American National Standards Institute.

[28] Norsam Technologies. *HD-Rosetta Data Preservation*, January 2004. URL http://www.norsam.com/hdrosetta.htm, Date of download: 16th of January 2004.

[29] OCLC/RLG Working Group on Preservation Metadata, Online Computer Library Center. *A Metadata Framework to Support the Preservation of Digital Objects*. Dublin, Ohio, June 2002. URL http://www.oclc.org/-research/projects/pmwg, Date of download: 3rd of March 2004.

[30] Ogle, V., and Wilensky, R. Testbed development for the berkeley digital library project. *D-LIB Magazine* (July/August 1996). URL http://www.dlib.org, Date of download: 13th of December 2003.

[31] Oltmans, E., van Diessen, R. J., and van Wijngaarden, H. Preservation functionality in a digital archive. In *Proceedings of the Joint Conference on Digital Libraries, June 7-11, Tuscon, Arizona, USA* (2004), ACM, pp. 279–286.

[32] Preserving Access to Digital Information (PADI) Migration. Website, December 2003. URL http://www.nla.gov.au/padi/topics/21.html, Date of download: 20th of December 2003.

[33] Patkus, B. *Assessing preservation needs, a self-survey guide*. Northeast Document Preservation Center, Andover Massachusetts, 2003. URL http://www.nedcc.org, Date of download: 15th of November 2003.

[34] Pettitt, J. *PRONOM - Field Descriptions.* The National Archives, Digital Preservation Department, December 2003. URL http://www.records.pro.gov.uk/-pronom, Date of download: 14th of Mai 2004.

[35] Philips, M. PANDORA, Autralia's Web Archive: An overview, 2003. URL http://pandora.nla.gov.au/background.html, Date of download: 3rd of September 2004.

[36] Rauch, C. Digital testbed for preservation strategies. URL http://www.ifs.tuwien-.ac.at/rauch, Date of download: 12th of April 2004.

[37] Rauch, C., and Rauber, A. Preserving digital media: Towards a preservation solution evaluation metric. In *Proceedings of the International Conference of Asian Digital Libraries* (2004).

[38] Rauch, W. Digital Immortality or Age of Oblivion? Contribution to the Working Group 'Digital Immortality and its Limits' at the European Forum Alpbach on the 27th of August 2004.

[39] Research Library Group, 2004. URL http://www.rlg.org, Date of download: 3rd of September 2004.

[40] retroK, and Azrail. aep Emulation page, 2004. URL http://www.aep-emu.de, Date of download: 8th of September 2004.

[41] Ross, S., and Gow, A. *Digital Archaeology: Rescuing Neglected and Damaged Data Sources.* Library Information Technology Centre, South Bank University, London, United Kingdom, 1999.

[42] Rothenberg, J. *Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation.* Council on Library and Information Resources Washington D.C., 1999. URL http://www.clir.org/pubs/reports/rothen-berg/contents.html, Date of download: 12th of December 2003.

[43] Rothenberg, J. *Carrying Authentic, Understandable and Usable Digital Records Through Time.* RAND Europe, Leiden, The Netherlands, 1999. Report to the Dutch National Archives and Ministry of the Interior.

[44] Schüller, D. Digitisation - The Only Viable Way to Preserve Audio Recordings in the Long Term. Presentation at the ERPANET Workshop in Vienna, May 11th 2004.

[45] Slats, J. Practical Experiences of the Digital Preservation Testbed, Office Formats. Presentation at the ERPANET Workshop in Vienna, May 11th 2004.

[46] Stinson, D., Ameli, F., and Zaino, N. *Lifetime of KODAK Writable CD and Photo CD Media*. KODAK, Digital & Applied Imaging, Rochester, 14650 New York State. URL http://www.cd-info.com/CDIC/Technology/CD-R/Media/Kodak.html, Date of download: 12th of March 2004.

[47] UNESCO, Information Society Division. *Guidelines for the preservation of digital heritage*, October 2003. URL http://www.unesco.org/webworld/mdm, Date of download: 05th of October 2003.

[48] Waugh, A., Wilkinson, R., Hills, B., and Dell'oro, J. Preserving digital information forever. *Digital Libraries* (2000).

[49] Weirich, P. *Decision Space: Multidimensional Utility Analysis*. Cambridge University Press, 2001.

[50] Wheatley, P. Migration - A CAMILEON Discussion Paper. *Ariadne 29* (2001). URL http://www.ariadne.ac.uk, Date of download: 12th of March 2004.