FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Content Profiling for Digital Preservation

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Petar Petrov
Matrikelnummer 0508142

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber, Univ. Doz.
Mitwirkung: Dr. Christoph Becker

Wien, TT.MM.JJJJ

(Unterschrift Petar Petrov) (Unterschrift Betreuung)

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Content Profiling for Digital Preservation

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Petar Petrov
Registration Number 0508142

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber, Univ. Doz.
Assistance: Dr. Christoph Becker

Vienna, TT.MM.JJJJ

(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Petar Petrov
Kochgasse 34, 1080 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)	(Unterschrift Petar Petrov)

Acknowledgements

Part of the following work was supported by the European Union in the 7th Framework Program, IST, through the SCAPE project, Contract 270137.

I would like to thank my supervisor, Prof. Andreas Rauber, for the insightful and inspirational conversations, the helpful comments and constructive suggestions, as well as for giving me the opportunity to work with one of the best people in the digital preservation community and letting me be part of his team and the SCAPE project.

I also would like to thank my other supervisor and reviewer, Dr. Christoph Becker, for his guidance, technical support and suggestions during the creation of this document and the prototype implemented as part of this work. Without him, everything presented here would have never been a part of the SCAPE project and I probably wouldn't have written it, as he inspired me and encouraged me to do so. I thank him for all those critical, but constructive reviews and talks that sometimes made my life seem hard but without which this thesis wouldn't be. I also thank him for helping me disseminate this work and supporting me throughout many long meetings and heated discussions with other partners within the project.

Last but not least, I would like to thank my family not only for supporting my studies abroad financially, but also for providing me with good example and for always being there for me offering help and good advice.

Thank You!

Abstract

Information Technology enables us to organise and manage our digital content into collections in an easy fashion. As a result, massive volumes of data are produced each day. However, it creates a huge set of technical and social issues regarding its safety and long-term accessibility.

Digital Preservation copes with issues related to hardware and software obsolescence and tries to keep our digital content accessible in the long term.

In order to make a meaningful decision about the course of action that should be chosen for a digital collection, preservation planning is conducted. The result of this rather complex and time-consuming process is a preservation plan. A preservation plan is an artefact that specifies a concrete action for the preservation of a set of objects and includes potential alternative actions and the reasons for the decision-making. The decision is based on knowledge about the content and the evaluation results of experiments conducted over sample objects of the collection. Inarguably, a content profile which is a thorough description of the collection and a small set of representative sample objects, is crucial for effective planning.

In general, the content profiling process consists of three parts; characterisation, aggregation and analysis. Characterisation is responsible for the extraction of meta data and the identification of digital objects, while aggregation offers a compressed view on them. In the last step of analysis, relevant aspects of the content are found and presented for further processing by preservation planning.

Because of the large volume of data, planners face many technical challenges. On the one hand, characterising millions of digital objects is a cumbersome and error prone process. On the other hand, aggregating output of various characterisation tools with complex output schemas is a highly tedious task that requires the expertise of preservation experts and is almost impossible on large scales. The lack of a thorough description and overview of the data often forces planners to select sample objects at random or based on a single property of the data. This results in subsets that are not representative and could lead to biased experiments.

The current state of the art does not offer solutions that are able to automatically create an in-depth profile of a significantly large set of digital objects, select representative samples and expose them in a semi-structured format.

In this thesis we observe the existing gaps in terms of content profiling and its importance within preservation planning. The contribution of this work is a conceptual solution of the content profiling problem, how it could be approached and a software prototype implementing the process. The presented prototype can operate on collections of about a million objects in scale. It helps to conduct an in-depth analysis, as well as select sample objects based on different algorithms. We evaluate the prototype using data collections of significant size in two case studies.

Kurzfassung

Informationstechnologien helfen uns, unsere digitalen Inhalte leicht zu verwalten. Dies ist der Grund für die zur Zeit bemerkenswerte digitale Datenproduktion. Allerdings, werden dadurch viele technische und soziale Probleme, die mit der Sicherheit, Langzeitarchivierung und Zugriff zu tun haben, verursacht.

Digitale Langzeitarchivierung versucht genau diese Probleme zu lösen, die mit Hardware- und Softwareveralterung zu tun haben, sowie auch den zukünftigen Zugriff zu garantieren.

Um eine sinnvolle Entscheidung über die Zukunft von einer digitalen Kollektion zu treffen, muss man einen Planungsprozess befolgen. Das Ergebnis von diesem komplexen Prozess ist ein Langzeitarchivierungsplan. Dieser ist ein Artefakt, das die konkrete Aktionen für die Langzeitarchivierung von einer Menge von digitalen Objekten spezifiziert und potentielle Alternativen und Gründe für die getroffene Entscheidung umfasst. Die Entscheidung basiert auf Wissen über die Inhalte und auf die Ergebnisse von Evaluirungsexperimenten, die auf Beispielobjekte durchgeführt werden. Aus diesem Grund ist die Erstellung eines Content Profile, das aus einer umfassenden Beschreibung der Kollektion, sowie aus einer kleinen Menge von Beispielobjekten besteht, unbestreitbar entscheidend für einen effektiven Plannungsprozess.

Generell besteht Content Profiling aus drei Teilen: Charakterisierung, Aggregation und Analyse. Im ersten Schritt wird eine Identifikation der digitalen Objekten durchgeführt und Meta Daten werden extrahiert. In der Aggregationsphase werden die gesammelten Daten in einer kompressierten Form dargestellt. Im letzten Schritt werden relevante Aspekte der Kollektion durch eine tiefgehende Analyse festgestellt und für Weiterverabeitung bereitgestellt.

Experten stehen heutzutage wegen des großen Volumens von Daten vor vielen Herausforderungen. Einerseits ist die Charakterisierung von digitalen Objekten ein umständlicher und fehlerhafter Prozess. Andererseits ist die Aggregation von den Ausgaben unterschiedlichen Werkzeugen mit komplexen Schemata eine Aufgabe, die Fachkenntnisse von Experten braucht und die schwerfällig auf großen Skalen ist. Das Fehlen einer umfassenden Beschreibung und Überblick sind oft der Grund für die Auswahl von Zufallsobjekten. Dies führt zur Selektion von Objekten, die nicht repräsentativ sind und kann zur gefälschten Experimenten führen.

Nach dem aktuellen Stand der Langzeitarchivierung existieren keine Lösungen, die es erlauben einen detaillierten Profil von signifikanten Datensätzen automatisch zu erstellen, repräsentative Teilmengen auszuwählen und in einem semi-strukturierten Format darzustellen.

In dieser Arbeit betrachten wir die existierenden Lücken im Bereich des Content Profiling und des Planungprozess. Der Beitrag dieser Arbeit besteht darin, eine konzeptionelle Lösung des Problems sowie eine Implementierung in Form eines Prototypen zu erstellen. Der Prototyp kann auf Kollektionen von substantieller Größe arbeiten und hilft bei der tiefgehenden Analyse. Abschließend wird dieser Prototyp anhand zweier Fallstudien von Datenkollektionen mit signifikanten Volumen evaluiert.

Contents

CHAPTER 1

Introduction

This chapter introduces the problem of preserving digital information and gives some perspective of the volume of data we have to deal with. It also gives a motivation for this thesis as well as a summary of the problem of content profiling. Afterwards the aim of this work and how the problem was approached is outlined.

1.1 Content & Digital Preservation

The information age enables us to produce, transfer and share massive volumes of data freely in an easy fashion. Scientists all over the world conduct complex research experiments and simulations that produce such an enormous amount of information that was unimaginable just a couple of decades ago. Jim Gray gives a simple example of the order of magnitude of data volume growth in the near future. The Large Synoptic Survey Telescope[1] will take snapshots of the entire night sky every few nights. Its construction start is scheduled for 2014 and full operations will start in 2022. It will produce around 1.3 petabytes of information only in its first year of operation, which is far more data than any other telescope in history has produced [18]. This particular example demonstrates how technology influences digital data production in scales that become harder and harder to manage.

The World Wide Web has certainly played an important role as a catalyst of this data growth. In order to put this fact into perspective, Intel®[2] has created a famous Infographic presented in figure 1.1. One breathtaking example is that in 2015 it will take you 5 years to watch all video materials that cross the IP networks each second. As it becomes easier and cheaper to create, edit, manipulate, store and share large amounts of digital objects, people often grow unaware of the problems that arise with the digital content they create.

A single sheet of paper, put in a normal environment, can easily endure a number of decades and will most likely still be readable and accessible and even semantically understandable. A

[1] http://www.lsst.org/lsst/about
[2] http://www.intel.com/

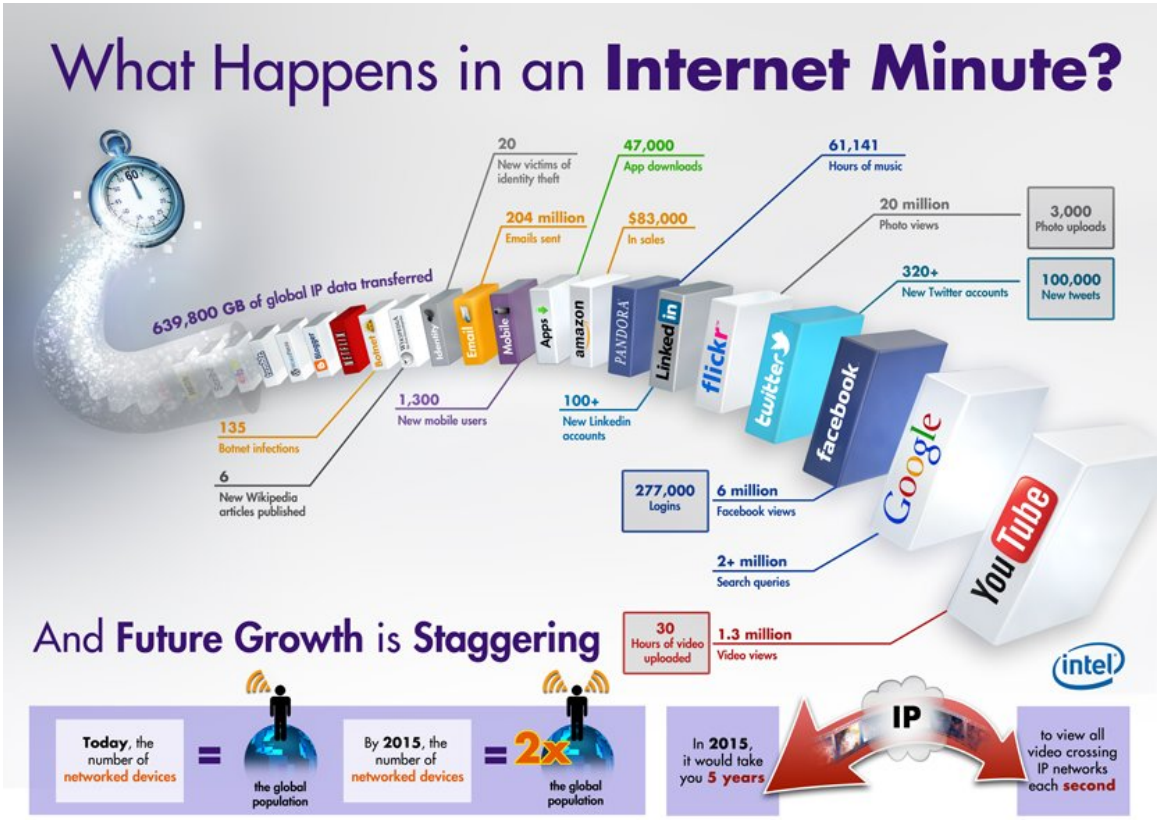

Figure 1.1: What happens in an internet minute. (An infographic by Intel©)

digital object, a file that contains the exact same content, often does not stand a chance of living through the next decade. Hardware failures, software obsolescence, changed environments, lack of backup copies are just a small set of examples of what may occur to digital objects and render a user unable to access them again.

> *Digital preservation refers to the series of managed activities necessary to ensure continued access to digital materials for as long as necessary. Digital preservation . . . refers to all of the actions required to maintain access to digital materials beyond the limits of media failure or technological change. Those materials may be records created during the day-to-day business of an organisation; "born-digital" materials created for a specific purpose (e.g. teaching resources); or the products of digitisation projects* [7].

It is aiming to preserve digital content through the years and make it findable, accessible, readable and understandable for periods of time which often surpass the lifetime of hardware and software components [40].

In the last years, a growing awareness of digital preservation problems is seen throughout scientific communities, memory institutions and business enterprises. These create solutions and follow different approaches and apply specific workflows on content in order to tackle many problems on different levels.

Currently, content holding institutions possess huge amounts of data. Web Archives, for example, crawl and store web sites and related content, such as images, video, style sheet files, etc. National and state archives use digital repositories to preserve their content. However, crawling and storing the bits and bytes represents only one step towards preserving all this content.

Regardless of the origin of the content (web archive, scientific data, personal audio and video collections, etc.), preservation planning has to be done in order to be able to identify and apply the most meaningful course of action and thus ensure the long-term safety and accessibility of each object that was stored. And since content and the environments that we use to manage it are continuously evolving, this process has to be repeated on a regular basis.

1.2 Motivation

Due to the fast growth and scale of data, the need of automation support in digital preservation processes arises. In order to conduct preservation planning effectively, one has to undertake a well-defined process consisting of numerous steps. First, a detailed definition of the requirements is created. Here the scope of the plan is defined by creating a description of the content and the requirements, as well as by selecting a small subset of sample objects. In the next step, different experiments are conducted over the samples and the results are evaluated. Finally, the results get analysed objectively and a preservation recommendation is created [2]. As the description of the collection and the selection of representative sample objects form the foundation of the plan, these parts of the process are considered to be very important.

The validity and effectiveness of the planning process is highly dependent on some analysis steps conducted by the planner. A definition of the content in terms of its meta data and properties is crucial. This meta data plays an important role when preserving digital objects, as it not only carries important information about the objects themselves, but also about their structure. The analysis of such information provides hints about the digital objects and their type and helps experts decide the best course of action regarding the long-term accessibility and preservation of data. Then a planner has to select a set of sample objects that are representative to the whole collection of objects. The careful and systematical selection of such objects forms the foundation of the experiments with reduced bias. A planner then decides the course of action by evaluating different options, called preservation alternatives. This is done with the help of experiments conducted over the chosen sample objects. Based on the results of these evaluation experiments, a planner chooses one of the alternatives. All this means that enough valid meta information and knowledge about the content has to be obtained, so that only proper alternatives can be chosen. In order to achieve this goal, some very important and necessary steps that provide valuable input to planning have to be taken. These are currently not always done efficiently, if done at all. For example, the content has to be characterised and all the meta data that is extracted has to be aggregated and analysed before all potential preservation action alternatives

can be found and evaluated. Based on this analysis, the content that is to be preserved can be split up into homogeneous sub-collections with similar characteristics and significant properties. Furthermore, such an aggregation of the similarities between different parts of the content can enable more efficient stratification of representative samples in an automatic fashion.

The preservation planning process is currently done semi-automatically and needs much input regarding the content profile from a preservation expert. Due to the scale of the data, a preservation expert often does not have a specific overview of the content, but rather high-level knowledge. This is the reason, why experts currently consider only high-level assertions as the profile of a collection. It seems that many parts of this process, from characterisation to plan deployment and execution, can be automated or at least enhanced by machine processes. Having an automated approach to create a machine-readable content profile, as well as support for scalable in-depth analysis of meta data will highly influence the performance and efficiency of planners. Becker, et al. identify scalability as one of the current challenges in a digital preservation environment. In order to achieve the desired scalability all involved processes have to scale up and make use of distributed architectures and state of the art algorithms [5]. If this is accomplished, the effect will be much more valuable than the sum of its parts for the Digital Preservation community.

Nonetheless, the current state of the art does not specify a concrete and well-defined way of how content profiling should be done and what information it should aggregate. What is more, there are almost no evaluation possibilities of the results that characterisation tools offer. Some tools try to give a confidence level, which is a first step towards such a validation, however it is not enough. In order to illustrate the problem, consider the ”character set encoding” of html files. Many files specify a UTF-8 encoding, however the real encoding used in the file is different. Some tools are able to detect this, whereas others just report the declared encoding. This is only one of many examples of such uncertainties that arise due to missing quality assurance steps during the characterisation process. Such seemingly minor peculiarities can cause huge impact on the chosen preservation alternative and thus on the final result of the preserved content. The ability to detect them on a larger scale, to find out subtleties and nuances between different objects in a collection and to select valid representative objects, will greatly enhance the preservation planning process.

Thus, a specific way of aggregating large amounts of meta data and its multi-dimensional representation would be of great value. Only with such a content profile can efficient preservation planning be achieved.

1.3 Problem Statement

Content Profiling is the process of obtaining an overview of a set of digital objects. The term ”content” does not refer to the information or semantics encoded into the object, but to descriptive and technical properties such as size, format, etc. which help understand the differences and similarities of the digital objects set. The meta data includes any technical property of the content that is considered relevant for preservation or the planning process. The profiling process usually results in a description of the meta data and includes representative sample objects that are used to conduct experiments.

Very often content profiling is done on a very high level that is often insufficient for a planning process, especially if this process is to be fully automated. The current state of the art of content profiling in preservation plans is creating a short description, written by a preservation expert. These descriptions are usually very high-level assertions, such as: “The collection contains about 2 million TIFF files”. The profile also contains simple metrics, such as (approximate) collection size, and number of elements as well as the formats identified in the collection. These measurements are important, but are not the only ones needed for the creation of an efficient preservation plan. For example, the information about the size of the whole collection is not enough. Other size related measures, such as the overall size of all files in a collection that have a specific format, or the average size of files with mime-type 'text/plain', can be much more significant and helpful in a preservation planning process. The formats within a collection are another example where simple measures are not enough. In cases of heterogeneous content, the distribution of the formats in combination of some other property can turn out to be very helpful for analysis and comprehension of the content. On the other hand, in cases of format-homogeneous collections the file format alone will play only a minor role. Combining the format with other properties such as the creation date or the creating application could however give deeper insight into the collection. These are only a few of the examples of multi-dimensional characteristics that could turn out to be important in content profiling. However, there are many more that will play a different role from case to case.

Furthermore, the creation of a plan has very specific requirements. For example, it needs a small subset of the collection in order to conduct some experiments over it. Based on the results, recommendations and decisions about the preservation actions of the whole collection are produced. Thus the choice of a representative collection subset is a very important process that is unfortunately often taken lightly, e.g. done randomly or done based on a very shallow analysis. Besides of the manual and random choice of samples in current preservation practices, their integration within the planning process is also done manually (either by file upload or manual filling of the related sample information).

Currently, there is no tool that meets these planning requirements and tries to tackle and solve problems arising from the large volume of content, the sparse meta data, the conflicts in the measurements and many other.

The rapid data growth introduces even more problems. In the case of web archives, for example, much work has to be done in capturing the significant properties of such dynamic content as the World Wide Web, since the harvested content gets outdated almost immediately after the crawl has been done.

1.4 Aim Of The Work

In this thesis we aim to analyse the requirements for such a content profiling process and to create a well-defined approach to generate and represent it, so that it can be used as input to other tools, e.g. preservation planning and preservation monitor components.

This will provide a strong basis for preservation planning experts and processes as well as set the foundation for experiments with reduced bias.

The architecture of a software tool that is able to read the characteristics of large amount of files and generate a content profile is to be designed as well as a prototype to be implemented. The prototype should have well-defined extension points and interfaces so that integration with other digital preservation information systems is possible. This is expected to create much value to preservation experts.

1.5 Methodical Approach

In a first step a research of the current methods in scientific projects and institutional practices regarding content profiling is done. Based on this, a short analysis of the existing gap between the idea of content profiling and the actual steps done is carried out.

The main part of the thesis is the creation of a specification and the architecture of a software tool that is able to profile larger amounts of data and produce output conforming to that specification. In the next step, research about applicable algorithms able to find a (representative) subset of a given collection of characterised digital objects is performed, and a prototype implementation is created for the presented content profiling approach.

Afterwards two case studies are conducted, where the produced prototype is applied on a the Open GovDocs1 collection provided by the Digital Corpora for scientific experiments, which consists of nearly one million files and has a total of almost half a terabyte, and on real web archive data with a similar volume, provided by the Danish State University Library.

In the last step an evaluation of the tool and the methods applied is done and based on it a conclusion is drawn.

1.6 Structure Of The Work

This work is structured as follows: The next chapter offers an overview of digital preservation and preservation planning. It also summarises the state of the art of content profiling. At the end the author presents some observations about the current related work, based on an example scenario. In Chapter 3, a theoretical view of content profiling is presented, where its requirements, issues and open challenges are discussed. Chapter 4 presents the architecture and gives deeper insight into a prototype framework that is designed and implemented as part of this work. The following Chapter 5 describes the case studies that were conducted in order to evaluate the tool and draws conclusions about the content profiling approach proposed in this thesis and its implementation. In the last chapter a summary of this thesis is provided as well as the open issues and next steps regarding content profiling that have to be undertaken in future research.

CHAPTER 2

Digital Preservation & Preservation Planning

This chapter discusses related work in the field of digital preservation. First some general information about projects and activities are presented, and then an overview of the most common techniques of dealing with digital content in the long-term are outlined. Then a definition of the term collection is given and important aspects such as meta data analysis and scalability are discussed. Subsequently the chapter gives an overview of the current state of the art in tool support and discusses the influence and importance of quality assurance of measurements with respect to content profiling. At the end, we draw observations about the state of the art and the identified problems and gaps.

2.1 Preservation

More and more information is produced in digital form and has only a digital representation. This has enormous implications for national and state archives, libraries, scientific institutions and business enterprises, but also the small companies and even private people, as they often face data corruption and access problems in the long term. In general, digital preservation (or DP for short) copes with two main problems; preserving content (bit streams) for longer periods of time, and ensuring these contents are accessible and understandable in the future. When talking about "the future" or "longer periods of time", informally we mean: "as long as the content is needed". This abstract definition of "Time" and "Long Term" can be more formally defined by the time needed for a changing technology to have a considerable impact on the data or the time for the requirements of the user community to change. This time span can be indefinitely long. [9]. That is the reason why the field of DP is full of challenges. These span from the fundamental technical problems through organisational and social issues to practical and financial ones.

A good example to picture the problem and challenges in DP is presented in [28] and in [36]. Imagine a file created today on a specific physical machine. This file is nothing more than a series

of bits shaped in a specific format. In order to access this file in the long term, not only the bits and bytes have to be preserved but also the way of interpreting them (the format specification). This would also require preserving the programs that can open, render and manipulate the file, which in turn will require the preservation of the dependency libraries and software packages as well as the operating system and the whole environment in which these programs or program versions run. Failing to preserve only one single part of this chain, the content would be lost (even if the physical bit stream is still in tact). Heslop et al. look at this problem from a different angle and introduce the term "performance". The distinction between an article or newspaper in its paper form and its digital record representation is that the digital record is the result of the interaction between technology and data as opposed to the physical counterpart. This implies that the experience of the digital object only lasts during this interaction and that each rendering of the record is actually a new original copy of itself. If two or more agents render the object they should experience equivalent performances of this particular record. As a result the term 'original' does no longer refer to the original paper document but to the original performance of the particular record on a screen or a device where it was viewed [17].

Due to this and many other problems, a community of preservation experts has emerged. Through the last decade a number of DP-related research projects and initiatives have been established. These have identified problems and threats and have advanced the state of the art in this filed. Starting in the mid nineties, scientists recognised that these problems could lead to disasters and thus the need of digital preservation and its importance. By the beginning of the new millennium there were the first initiatives and projects in the EU that started focusing on research topics related to DP. These projects (ERPANET[1], DELOS[2], DPE[3]) were aiming the establishment of a community, identification of target groups and transfer of expertise. The first scientific research was focused on topics such as standards, system concepts, selection and appraisal policies and format identification. Ioannidis et al. suggested a reference model for digital library management systems describing the characteristics of such information systems in [21]. Rauber et al. defined a testbed framework for documenting the behaviour and functionality of digital objects and preservation strategies. The main goal of the framework was to find possibilities to automate preservation experiments [37]. All this set the foundation for more technical and practical approaches that were undertaken in later projects (PLANETS[4], CASPAR[5]). The aim of these was to research the preservation of simple digital objects such as office documents and images. These projects advanced the state of the art of digital preservation and, as a result, different tools and languages that automated important DP processes were developed. PreScan, for example, maintains and preserves objects' meta data in an automated fashion [31]. Becker et al. proposed a generic XML language for characterising objects to support digital preservation. The language aimed to support automatic validation of document conversions by decomposing the documents structure and representing it in a generic XML language [6]. All this helped the establishment of a solid community and a body of expertise. An overview of the EU DP projects

[1] http://www.erpanet.org/
[2] http://www.delos.info/
[3] http://www.digitalpreservationeurope.eu/
[4] http://www.planets-project.eu
[5] http://www.casparpreserves.eu

and activities is presented in [44].

Present initiatives include more fundamental research that tries to focus rather on more complex and interactive objects than simple documents and data structures. Projects such as LiWA[6] attempt to solve issues related to Web Archiving, whereas projects such as TIMBUS[7] and WF4Ever[8] focus on the preservation of business processes and scientific workflows. Galushka et al. outline a complete framework for preserving a business process with all its relevant aspects and dependencies in [15]. Other projects such as SCAPE[9] build upon the solid framework established in the past and aim to improve the state of the art of DP by developing infrastructure and tools for scalable preservation actions and integrating them with automated policy-based preservation planning and preservation watch systems and workflows. Schmidt presents the design of a preservation platform architecture that supports distributed algorithms and fits into various digital preservation use cases in [42]. Faria et al. present the architecture of a novel preservation watch system that monitors specific preservation related properties of different sources and notifies interested parties when important or critical events occur in [12].

Common Techniques

Through the years many tools and procedures were developed in order to preserve digital content. In the literature there are often different names for the same or similar concepts. Here we present the most prominent ones and try to differentiate them and put them in perspective. There are different levels of concern: physical, logical and semantical. The physical level of concern deals with the data integrity, or in other words the correct storage of the bits and bytes. The logical preservation deals with problems such as their structure, formats, etc., and the semantical level copes with the problems of preserving the meaning of the encoded information. As the focus of this thesis is not much related to the semantical level, we will concentrate on the first two.

Physical Preservation

Migration is the technique of copying, moving or converting some source of information to another target. In the sense of physical preservation, it is the concept of moving the bits to a different medium with a different (physical) location. There are many different media that can store digital data. Some are more stable than others; some are more popular than others. No matter what type of medium is chosen for data storage (CDs, DVDs, Hard Drives, etc.), it is not guaranteed that the data stream is safe. Through physical damage, bit rot or other disasters, there is a high chance that your digital storage media will fail to reproduce your bit stream. Thus on this lower level the only option would be to copy the streams to a different medium from time to time. This is strategy is also often referred to as *refreshing* [27].

However, refreshing the data does not guarantee that it will be accessible in a later point in time, as new media are also error prone. Therefore, approaches like LOCKSS (Lots Of Copies Keep Stuff Safe) [38] make use of the distribution of many independent copies. Developed at

[6] http://www.liwa-project.eu

[7] http://timbusproject.net

[8] http://www.wf4ever-project.org

[9] http://scape-project.eu

the Stanford University, the LOCKSS approach was implemented in a librarian software system that deploys many low cost copies of persistent web-caches and enables the detection and repair of damages based on voting in opinion polls [30]. Following a LOCKSS approach, however, only minimises the risk of losing data. If there is no effort spent in management of the copies, then it is fairly easy to lose track of the copies. For a software tool this might seem irrelevant, but for a private user this is a real issue.

Furthermore, even if enough, well-managed copies have been stored and the data stream has been preserved, there is always the issue of software obsolescence and thus failure in the access and interpretation of the stream. A good example of this problem can be found in this blog post[10]) by the former head of the UK's Digital Curation Centre[11] Chris Rusbridge, who struggled to open and successfully read almost 15 years old PowerPoint presentation slides.

Logical Preservation tries to cope with the structural integrity of a digital object and how the data should be interpreted in order to render the encoded information of an object. It tries to deal with the problems of old and outdated formats and software. In order to preserve not only the bit stream, but also to ensure the integrity of a digital object and its successful interpretation in the long-term, another type of migration is often used [27]. New operating systems, new software tools or new versions are sometimes incompatible or unable to render and manipulate older formats. To cope with technology changes, digital preservation often uses a conversion strategy where the data is migrated (moved) to another format that is usually considered to be more stable than the original. A format is usually considered worthy and stable for preservation purposes when it is standardised, the format specification is open and well documented and there are no patent owners and license fees that apply. The Florida Center for Library Automation, for example, offers a report[12] with the recommended data formats for preservation purposes that is considered as a good reference and starting point.

However, there is no ultimate reference table or no ultimate file format that fits all preservation purposes. From use case to use case, different aspects have to be considered. Neither standards, nor migration tools alone can ensure that a digital document remains accessible and its integrity remains unharmed. The format is only the tip of the iceberg, as the problem is various and manifold. In the end, there are always many different aspects that have to be considered and it always comes down to a multi-criteria decision making problem [4].

Wing and Ockerbloom further discuss the topic as they analyse what information is preserved by type converters and formalise the notion of respectful type converters in [48]. Informally, a migration tool (or a converter) respects a certain type T if an original object A and a converted object B show the same behaviour when viewed as objects of type T.

Rothenberg gives a good overview of common flaws of the concepts of migration in [41] and summarises important aspects that should be considered in DP. For example, the translation to a new format has the flaw that the original is usually discarded afterwards (e.g. because of storage costs). This can make it impossible to validate, whether or not information has been lost.

[10] http://unsustainableideas.wordpress.com/2012/10/15/ppt-4-adventure-learning/

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

[12] http://fclaweb.fcla.edu/uploads/recFormats.pdf

Nonetheless, format migration is often applied within digital preservation systems and repositories. As it also has potential pitfalls, it is not to be taken lightly. A more practical downside to migration is the storage cost. Often the target format has a bigger footprint than the original. Also the conversion of huge amounts of data is an error prone process that is not easy to validate [28]. Thus the originals are often kept for a certain period of time after the migration. Furthermore, if the migration path consists of several steps one, has to make sure that all required meta data of the original is also migrated to the new versions of the objects. Another related issue is also quality assurance. As it is infeasible to check manually if the conversion process was successful, there are very specific requirements and processes that have to be followed in order to automate the verification of the preservation action [4, 29]. All these and other issues have to be carefully taken into account before choosing such a preservation action.

Another common technique of logical preservation is emulation. It has the verb "emulate" in its root and means to imitate or reproduce. In software terms, an emulator is a software tool that imitates the behaviour of a (hardware) system/framework (usually an older one) in order to run other (obsolete) tools that are meant to run on the emulated system. Clearly, this approach can come in handy in some DP activities. In [41] Rothenberg gives an overview of a process for preservation that is based on an emulation process. The author states that not only the data (bitstream) has to be stored but also the bit stream of the original program, the operating system and all other necessary parts, e.g. dependencies and used libraries. Also a thorough and complete description and specification of the underlying architecture have to be provided in a form that is readable by potential future emulator authors. If these prerequisites are met, an emulator that mimics the specific hardware needed to run the tool can be created. Lorie points out in [28] that the specification of the architecture has to be perfect and complete, which is an immensely difficult task. Another very important argument he makes is the evaluation of such an emulator. Even if all needed input existed and a hypothetical emulator was created, how can its correctness be proven as no original hardware device exists?

These and other reasons combined form one of the biggest downsides of emulation; cost. The effort, manpower and infrastructure needed to emulate an environment that renders digital documents is often more expensive than the value of the content of the documents themselves.

Nevertheless, emulation is widely used in specific branches, such as video gaming. Guttenbrunner et al have evaluated different strategies for the preservation of console video games in [16] and came to the conclusion that while migration shows very good results for the preservation of visual and audio components, it completely fails in interactivity. Emulation, on the other hand, showed promising results. All of these, however, were strongly dependent on the sample objects that were emulated. Some of the evaluated emulation alternatives worked great on some of the sample records but completely failed on others, which is an important aspect to keep in mind when selecting representative objects and stresses the importance of the way these are selected.

Preservation Planning

Preservation planning is a key task in DP that has to be undertaken by every institution or person that is serious about preservation. Numerous DP related projects have investigated the key

requirements and processes involved in preservation planning. Projects such as PLANETS[13] have created very strong fundaments in this area and have developed tools such as PLATO[14] - the preservation planning tool which supports a standardised workflow and helps users throughout numerous steps with the goal of creating a preservation plan. Follow up projects, such as SCAPE[15] advance the state of the art and enhance the planning capabilities by improving the current status. This involves building up a framework around the process that supports many new features such as preservation monitoring services, semi-automatised experiments execution, automatic plan deployment and execution and many more enhancements to the whole process in order to provide a scalable, robust preservation planning process.

What is Preservation Planning?
The OAIS reference model was developed by the Consultative Committee for Space Data Systems and soon afterwards was accepted as an ISO standard [22]. This high-level reference model has proven to be a helpful tool for the DP community for many years. Undoubtedly, one of its key parts is Preservation Planning or PP for short.

At the core of PP is the recommendation of archival operations. This requires a decision making process that evaluates different preservation strategies or actions and chooses the most appropriate one. The outcome of the process is highly dependent on object characteristics and institutional settings and requirements [43]. The goal of the process is to create a preservation plan that documents all the steps and choices that were made, the policies that were followed while making these decisions and the different preservation alternatives that were evaluated. It offers a complete documentation of the decision that allows one to repeat all experiments and verify why a certain preservation action was chosen at a given point in time.

Figure 2.1 depicts the planning environment of the whole planning process. It constitutes of four main phases: define requirements, evaluate alternatives, analyse results and build preservation plan.

In the first step, all necessary requirements for the specific use case are collected. This includes the identifiers and a profile of the objects or the collection that is examined, but also the technical environment and specific usage criteria for the particular set of objects. Another important part are the organisational policies, which are usually high level statements that guide the organisation.

In the next step, different alternatives are evaluated. These are called preservation actions and implement different preservation strategies, depending on the use case. Depending on the chosen strategy, different preservation action tools can be evaluated by conducting different experiments over a set of objects. As every strategy offers a set of tools and every tool has its own set of configuration parameters, this step may become quite expensive in terms of time and resources.

As soon as the experiments are conducted, a planner has to analyse the results objectively and decide which of the tested actions makes most sense considering the requirements and the

[13] http://www.planets-project.eu/
[14] http://ifs.tuwien.ac.at/dp/plato
[15] http://www.scape-project.eu/

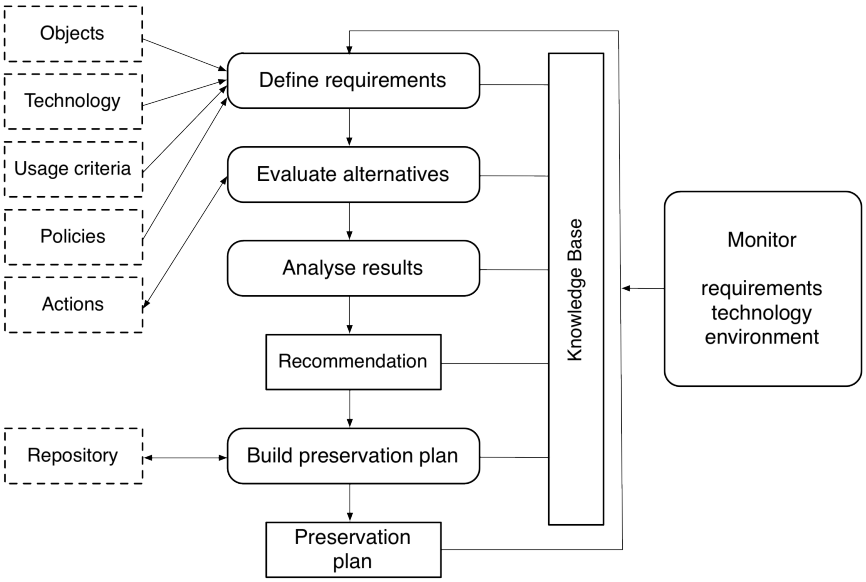

Figure 2.1: The preservation planning environment [2].

experiment results. It is important to note, that keeping the status quo is a perfectly valid solution in many use cases and is also supported by the planning process.

After a recommendation is made, a preservation plan is created, which is a very concrete artefact, as opposed to policy documents, that specifies an action plan for the preservation of a set of digital objects.

> *A preservation plan defines a series of preservation actions to be taken by a responsible institution due to an identified risk for a given set of digital objects or records (called collection). The Preservation Plan takes into account the preservation policies, legal obligations, organisational and technical constraints, user requirements and preservation goals and describes the preservation context, the evaluated preservation strategies and the resulting decision for one strategy, including the reasoning for the decision. It also specifies a series of steps or actions (called preservation action plan) along with responsibilities and rules and conditions for execution on the collection. Provided that the actions and their deployment as well as the technical environment allow it, this action plan is an executable workflow definition* [2].

Since planning is not a one-time activity, there is also an external monitoring phase that keeps track of important preservation related aspects of the world, such as new technology, formats and the organisational policies. It feeds back this important information about relevant changes in the environment, which can cause a reiteration/re-evaluation of a preservation plan. A detailed overview and a high level design of such an addition to the preservation planning environment can be found in [11].

As content profiling is a very important and necessary part for efficient and successful preservation planning, we take a look at the whole process in detail in order to understand where profiling fits in the process and how can it help enhance it later. It is a well-defined workflow consisting of four phases with several steps which are discussed in [43].

In the first phase, *"Define Requirements"*, the scope of the preservation plan is demarcated. The preservation expert has to follow three steps; to provide information about the collection, environment etc. (**Define Basis**), to choose representative sample records for experimentation (**Define Sample Records**) and to identify the requirements for the preservation plan or the so called objective tree, which summarised high-level goals of the plan (**Identify Requirements**).

In the second phase, *"Evaluate Alternatives"*, another 5 steps have to be followed. Starting with the definition of alternatives (**Define Alternatives**), the responsible preservation expert has to choose a set of potential actions, with all related information such as environment, tool invocation parameters, etc. In the following (**GO/NO-GO**) step a decisions is made whether to proceed or not based on each preservation action, the estimated resources and the defined requirements. After that the planner has to create suitable experiments (**Develop Experiment**), which are well-documented, repeatable set of actions with their environment and the capability to capture their results. In the following (**Run Experiment**) step, each preservation action is executed against the chosen sample records in order to obtain different results. In the last step of this phase (**Evaluate Results**) the results of the experiment output is evaluated against the objective tree in order to check if the identified requirements were met or not.

The third phase, *"Consider Results"*, is responsible for the objective analysis of the results. In its first step (**Transformed Measured Value**) all experiment results are transformed into the same scale (0-5) making use of special transformation tables and utility function. The following (**Set Importance Factor**) step provides the ability to equal the weight of different parts of the requirement objective tree, as not all goals are equally important. In the last step (**Analyse Results**) all measures are aggregated per objective and provide the planner with a preservation action recommendation and the necessary basis for a decision.

The workflow is depicted in figure 2.2 and provides the current state of the art in preservation planning matters. Although it provides a very solid theoretical ground and a complete specification that is working in practice, there is one part of the concept, which might be the cause of errors caused by human incompetence or lack of knowledge and understanding of the collection.

As one can see from the workflow, all the results strongly depend on the defined goals and the analysis of the experiments output. We assume that a preservation expert will understand the objectives of his organisation. Since the identification of requirements and the setting of the goals and objectives are very important steps that are also strongly dependent on the organisational background of the preservation expert, we also assume that it is unlikely they will cause errors and misunderstandings in later steps. Also if the planner happens to choose wrong preservation action alternatives, the result will be in the worst-case scenario a 'Do Nothing' alternative, which will not solve the problem at hand, but will also not do any damage. However, there is one step that could have serious implications and even cause damage or at least resource loss if taken lightly. Consider the following example, where the chosen sample records are picked up at random from a medium sized collection with several thousands of objects. Then

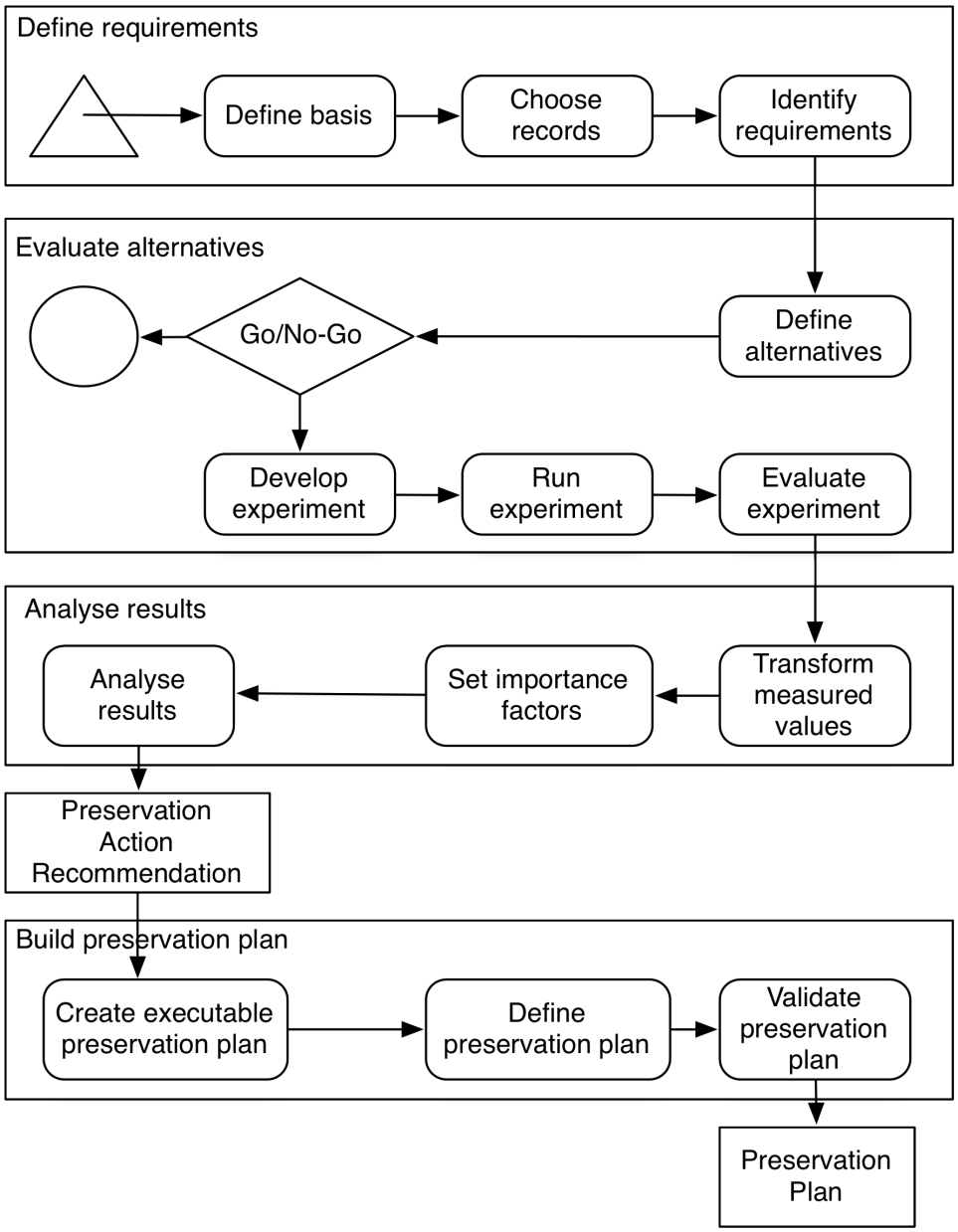

Figure 2.2: Overview of PLANETS Preservation Planning workflow [3].

the experiments show a particular preservation action is very feasible and the planner chooses to execute this action over the whole content, due to the experiments output and the consequent analysis. Although the analysis and the decision are perfectly valid (in their implementation and execution), it could turn out that the result of the preservation operation does not meet the requirements defined. This could happen, due to many different aspects in the format and content profile of the collection at hand. To summarise, the defined requirements and objectives are ok, the experiments are valid, the analysis is correct, but the overall results are not feasible due to the false premise, that the random chosen set of sample records is representative. Damage could be done, if there is no reasonable and thorough quality assurance process afterwards. This work addresses this problem and tries to prevent it by reducing the bias of the experiments. This ought to be achieved by thorough analysis and automatic representative selection in the early steps of the workflow.

One can argue, that no real preservation expert will choose representatives at random. Although this might be true, there are numerous other factors that have to be considered when choosing the representatives and since the collections that are worth preserving are often big enough, the overhead for the preservation expert is just not feasible to select them by hand. Thus the representatives are usually chosen by format and format version in combination with their size (minimum, maximum and average).

The planning tool PLATO, developed at the University of Technology in Vienna, implements this process and appends a fourth phase, where the user/planning expert can create an executable plan, which can be deployed within a repository. This fourth phase, (*"Build preservation plan"*) adds an important part to the workflow and results in an applicable real world preservation plan artefact. The preservation action plan is a well-defined specification that serves the purpose of documentation of the decision and contains the executable part, which specifies the tools, environment and parameters to use during the preservation operation.

However, in its current release, the planning tool supports only manual sample records definition. Although it assists the planner with integrated characterisation tools, it cannot provide higher certainty in the validity of the chosen representatives. Integration with another tool that provides a complete content profile (generated in an automatic fashion) would provide a huge benefit to preservation planners.

2.2 Content

As seen in the previous section, understanding the objects in a collection is a key part of the requirements for preservation planning. In order to create and analyse a collection or a set of digital objects, the meta data for each object has to be examined. Meta data is structured information about the data (objects) itself. It is usually stored within a file that provides further information about the content and format of the file as well as other important characteristics. In general, there are three main types of meta data: descriptive meta data for discovery and identification (e.g. title, author, etc.), structural meta data (page ordering, image width and height, etc.), and administrative meta data that helps management of the resource (e.g. creation date, type, etc.). The National Information Standards Organisation - NISO[16] has provided a

[16] http://niso.org

series of articles and reports in order to help people, archivists and experts understand meta data and its importance [34].

Today, analysis of such huge content often is done in a manual fashion, which can be a very time-consuming and cumbersome task. If there would be tools at hand that support identification and characterisation, data aggregation, filtering, collection splitting, etc., then the analysis process will be automated to a certain degree, hence analysis will be handled much faster and potentially much more efficient.

It is noteworthy that content here neither refers to the semantics of the information stored within the digital objects, nor to their visual representation characteristics or anything similar, but solely to the digital artefacts and the definition of the collection in terms of meta data, such as formats and format-related characteristics.

Consider a collection of digital scans of old newspapers. Its content here does not refer to the content of the newspaper articles, but to the meta data characteristics of the images that comprise the collection. For example, these may include, but are not necessarily restricted to the resolution of the scanned documents, their colour profiles, the size of the images or the number of pages. All these measurements play a huge role in the decision making process of preservation planning.

The following paragraphs provide more details on metadata and how this refers to content profiling for preservation planning.

Identification vs. Characterisation

Identification is the process of determining the format of certain sequence of bits and bytes. It is an important aspect not only for DP, but also in many other fields of computer science. Thus, there are numerous different methods of identifying the format of a file.

A trivial and popular approach is based on the extension of a file, or in other words the end of the file name. Many (early) versions of different mainstream operating systems use this approach. Unfortunately, it is rather flawed and unfit for DP for a number of reasons. For one, software or user clients easily manipulate the extension. Also, many files do not have an extension. Another problem is, that sometimes file formats produced by different software applications have the same extension.

A more sophisticated method for identifying a format of a file is by using its internal meta data. For example, this can be done by examining the file header (the beginning of the byte stream of a file) information. Many formats start with a few special bytes (called magic numbers) which define the following format [35]. Some tools make use of magic number tables and identify the format by comparing its magic number and table. Magic Numbers are a better solution than file name extensions, as they are harder to manipulate. The problem is that often magic numbers are not well documented and are easily lost. Furthermore, the tables used for comparison have to be maintained.

Characterisation, on the other hand, is the process of extracting more meta data out of the file, that can help explain the file, provide related information, such as the language, author, creating date, etc. Based on the different measurements of properties/characteristics a collection can be dubbed homogeneous or heterogeneous. Usually to a user a homogeneous collection would be a set of files that consists only or mostly of objects having the same format or even the

same type (audio, video, text, etc.). This, however, is an oversimplification, which has enormous side effects for preservation planning.

Consider a collection of N digital objects, which share the same extension, e.g. 'pdf'. To a normal user, this would be a homogeneous collection. An advanced user, however, would know that the extension of a file does not really specify the format of the file and thus could assume that there are differences. One step further would be to conduct an identification process that looks for the specific file format and format version. Assume that in our example 95% of all files have the same *PDF* format and format version. Then this could be considered a homogeneous collection with respect to the format. In a next step, however, characterisation is conducted and now there are many more properties, such as *creating applications*, *encryption*, *password protection*, *tags*, etc. Regarding these characteristics, the same collection can be considered to be heterogeneous. Clearly, all of this is very important to preservation planning, since different preservation actions produce different results exactly because of big differences in the values of such characteristics.

Thus the question remains, how to identify such important properties that define the homogeneity of a collection? Following this train of thought, clearly the format is a very important characteristic. However, it does not cover all cases (as in the example) and many others are important.

Preservation Analysis

As collections in DP are often just too large for a human being to comprehend, the meta data provided by identification and characterisation has to be aggregated and analysed in some fashion. For this purpose different statistical information is used to understand and stratify the content into different homogeneous parts. Often simple statistical measures about the size, such as minimum, maximum, average, standard deviation, etc. provide meaningful information about the current content one has to deal with. Moreover, histograms and distributions of mime types, formats, format version and other properties help to see the bigger picture of the content that is to be preserved.

Once the bigger picture gets clearer, the collection can be divided into different (more) homogeneous parts, which will ease the decisions that have to be made regarding their future with respect to DP and PP. In order to achieve this stratification, filtering or so-called slicing and dicing has to be applied on the data. This would include the traversal of the raw data and splitting the content into sets.

Another important part would be finding representatives within the homogeneous content. These are very small sized subsets (usually in the order of tens of objects) that are somehow representative to the selected content or part of it. The representativeness can be determined based on the distribution of different characteristics or combinations thereof. These representative samples form a better-suited common ground for the experimentation phase of the preservation planning process. Finding such small subsets within homogeneous collections is potentially much easier than in heterogeneous context.

Automating the retrieval of such a set is not a trivial task, as there are numerous factors to consider. Pan et al. discuss the topic in [32] and propose an approach for finding representatives in a massive data set by building a distribution table of different content features. The presented

algorithm relies on an objective function that is not well specified, but gives a good overview of the complexity of the task.

The problem of representativeness will be investigated in later parts of this thesis. The next chapter defines some informal requirements that a sample object set has to meet. Chapter 4 presents a small set of simple algorithms that are able to find representative objects in a collection based on different criteria. The aim is to identify potential problems and allow future work to build upon these and create more complex and sophisticated algorithms.

Scalability

As discussed in Section 1.1, content growth nowadays has a tremendous pace. This fact has some serious implications on how information systems have to deal with it. Scalability does not only pose a problem related to volume and performance, but also to usability and presentation, automation and costs.

Looking at the growth of content within web archives, for example, and their projection the problem of vertical scalability becomes clear. Vertical scalability (i.e. installing machines with better performance) will not be able to solve the problem of storage and data management, not to mention the effective analysis of data.

Since this is a problem not only related to digital preservation but to information systems in general, there are many studies for algorithms, technologies and architectures that enable horizontal scalability, i.e. attaching more commodity machines and distributing the payload among them.

Driven by economies of scale, Cloud Computing has played an enormous role in this area by providing a large set of easy to use and access resources, such as hardware, development platforms and services [14, 45]. Distributed Platforms as a Service (PaaS), such as Amazon AWS[17] and Amazon S3[18], Google App Engine[19], Heroku[20], etc. have proven to be very performance- and cost-effective. Distributed approaches and algorithms, such as Google's Map Reduce [10] have found many applications in various fields of computer science.

Map Reduce is a fairly modern parallelisation algorithm for processing large data sets on certain kind of distributable problems. The framework can make use of a large amount of nodes for the computation. It takes as input a set of key/value pairs and produces a set of output key/value pairs. It typically consists of only two functions: map and reduce. In some cases a third finalize function can be used to do some further computation that needs all the results of the reduce steps.

The Map function takes a set of pairs of keys and values in the form of (k1, v1) and transforms them to a set of intermediate key/value pairs. The framework groups together the intermediate values by key and passes them for further processing to the Reduce function.

The Reduce function accepts an intermediate key and a list of values associated with that key and is responsible for computing the (partial) final result. The reduce function can be invoked many times by the framework and there is no guarantee that it will be run on the same node as

[17] http://aws.amazon.com

[18] http://aws.amazon.com/s3/

[19] https://developers.google.com/appengine/

[20] http://www.heroku.com

the Map function. This means it has to be idempotent and agnostic to external knowledge about the distribution.

This rather simple approach has been widely accepted by the OpenSource community and was implemented by the Apache Software Foundation in a library called Hadoop[21]. The possibility for integration with a BigTable-like Store [8] (HBase[22]) and a distributed file system (HDFS[23]) has enabled many companies to handle the big volumes of data they have. Google was using a similar architecture for its index construction, article clustering and statistical machine translation. Yahoo uses it for spam detection in its mail service and other big corporations use it for data mining, ad optimisation and more.

All this sets a solid foundation for in-depth analysis of meta data on larger scales, which would greatly help planners to automatise the content profiling and preservation planning processes. Furthermore, planners will benefit from the fact that such technology will increase performance and most likely provide more efficient results.

State of the Art

Observing the current state of art implies that analysis of preservation related data should be feasible and cost-effective. Nonetheless, preservation analysis tools nowadays often lack the ability to analyse content on a larger scale, or if they support it, there is a trade off in the analysis depth. This fact is due to two common worldviews. On the one hand, there is the popular belief that the format is the one property that matters for digital preservation [19] and that there is no real need to look at many different aspects. On the other hand, the volume of the data is so high, that even the deep characterisation meta data volume can be considerably large, which impends large-scale analysis. These observations of the current state of the art of preservation analysis approaches are sketched in Figure 2.3. The X axis depicts the depth of the analysis of tools and frameworks in terms of considered characteristics. The Y axis depicts the volume of the meta data used for the analysis. The figure illustrates the trade off between large volumes of data and the number of identification and characterisation properties that can be analysed.

Another fact that contributes to the problem is that digital repositories often provide only simple metrics such as the format profile - an aggregation of the formats in the repository and their absolute occurrences, ingest and creation date and in some cases the applications. Even though the repositories often have the means to characterise the data or even store it, there is no easy way to obtain a bigger picture about a certain collection. Often the only way to obtain more specific information that relates two or more characteristics out of a repository involves issuing complex queries directly to the database of the repository. This is not user friendly and feasible to a planner, but also requires the knowledge of system administrators.

Last but not least, selecting representative sample objects is usually done manually. The chosen samples are often stratified based on size and format [26]. This might be enough, but there are certainly cases where looking at other characteristics and the combinations between is more important.

[21] http://hadoop.apache.org

[22] http://hbase.apache.org

[23] http://hadoop.apache.org/docs/hdfs/r0.22.0/hdfs_design.html

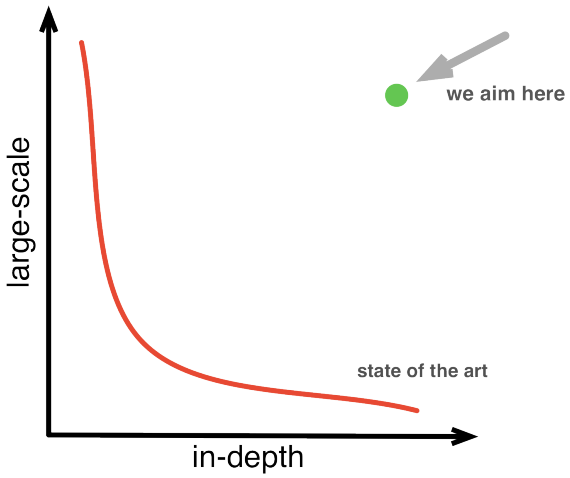

Figure 2.3: The current state of the art of preservation analysis tools and frameworks in relation to the volume of content and the depth of the analysis.

2.3 Tool Support

Due to the rising awareness of digital preservation and the numerous projects conducted in this area in recent years, many existing tools have found new use and many new tools were written from scratch in order to support DP activities. This section provides a short overview of some of the more prominent ones that are related to content profiling.

A recent report (created as part of the SCAPE project) summarises an evaluation framework and the results of the tests of several identification and characterisation tools [25]. It provides a rather good overview of the current state of the art of such tools but concentrates mostly on their identification capabilities. In the report six tools (DROID 6.0[24], FIDO[25], Unix File Tool[26], FITS[27] 0.5 and JHOVE2[28]) were evaluated against 22 criteria such as tool interface, its license type, platform dependencies, accuracy of reported results, documentation, etc. Another recent research conducted by the National Library of Australia has investigated four file identification tools (File Investigator Engine[29], Outside-In File ID[30], FIDO and file) and five metadata ex-

[24] http://sourceforge.net/projects/droid/
[25] https://github.com/openplanets/fido
[26] http://unixhelp.ed.ac.uk/CGI/man-cgi?file
[27] http://code.google.com/p/fits/
[28] https://bitbucket.org/jhove2/main/wiki/Home
[29] http://www.forensicinnovations.com/fiengine.html
[30] http://docs.oracle.com/cd/E16184_01/dev.837/e12875/title.htm

traction tools (File Investigator Engine, Exiftool[31], MediaInfo[32], pdfinfo[33] and Apache Tika[34]), some of which commercial. The results of these tests can be found in [20].

Here we summarise the strengths and weaknesses of some of these and other tools briefly in order to give an overview of the current state of the art.

- **DROID 6.0**
  Droid is an identification tool produced by the National Archives, which uses the PRONOM[35] registry and its format signatures and/or file extensions. It provides information about the mime type, format and format version of a file as long it is in the DROID signature file, which contains the 'magic numbers' of the PRONOM registry. It also outputs a PRONOM Unique Identifier or PUID, which can be used to trace the format back into the registry. Unfortunately, the registry is not open and its maintenance is slow. However, the tool is very useful and widely adopted within the DP community.

- **Apache TIKA**
  Tika is an open source project from The Apache Software Foundation that is able to extract metadata from files with various formats. It is a stable tool able to identify files by analysing their bitstream and allows the deeper characterisation of some of these files.

- **FIDO**
  FIDO is another identification tool that is a clone of Droid and also uses the PRONOM signature file registry. Although it has numerous glitches, FIDO's performance proves to be 35 times faster than DROID when working on one file at a time.

- **UNIX File Tool**
  File is a CLI utility application included in every Unix distribution and first released in 1973. The tool has stood the test of time and has proven to be very stable, which also makes it widely adopted in the DP community. It makes use of magic numbers to identify files and has been used in DP activities for a long time. It has a very good computational performance and supports large number of formats.

- **JHOVE**
  JHOVE is one of the most well known identification and characterisation tools used by the DP community. It is also developed by the Harvard University Library and is able to extract meta data from various formats based on different modules. Probably one of the most valuable features of JHOVE is the ability the check a file for well-formedness and validity against the format specification. Figure 2.4 shows a screenshot of JHove and the output of an example PDF file.

[31] http://owl.phy.queensu.ca/ phil/exiftool/
[32] http://mediainfo.sourceforge.net/en
[33] http://www.foolabs.com/xpdf/download.html
[34] http://tika.apache.org
[35] http://nationalarchives.gov.uk/pronom/

- **JHOVE 2**
  JHOVE 2 is a successor project for the JHOVE tool and also provides an extensible architecture for characterisation tools and modules. It is developed as an open source tool by the California Digital Library, Portico and Stanford University. Currently it produces helpful output only for a few types of documents as the different modules are not yet developed.

- **FITS**
  The File Information Tool Set is developed by the Harvard University Library. It wraps common identification and characterisation tools as the ones described here and tries to consolidate them and provide a normalised output. By providing basic provenance information for each extracted record, it combines the consolidation result and provides a very basic confidence status for the extracted value of each property. This proves to be helpful for cases where there are uncertainties. The framework is designed to be extended, so that other tools can be also added. The tool seems very helpful, although there are some instabilities and problematic cases. An example FITS output file is provided in Listing 2.1. Every FITS file consists of four sections: *identification*, *fileinfo*, *filestatus* and *metadata*. The 'identification' section provides information about the mime-type, format, format version and optionally external identifiers, such as PUIDs. The 'fileinfo' section lists some generic to the format characteristics, such as size and creation date. The 'filestatus' section gives information about the structure and the validity of the content according to the format. The last 'metadata' section lists all format specific characteristics for the characterised file.

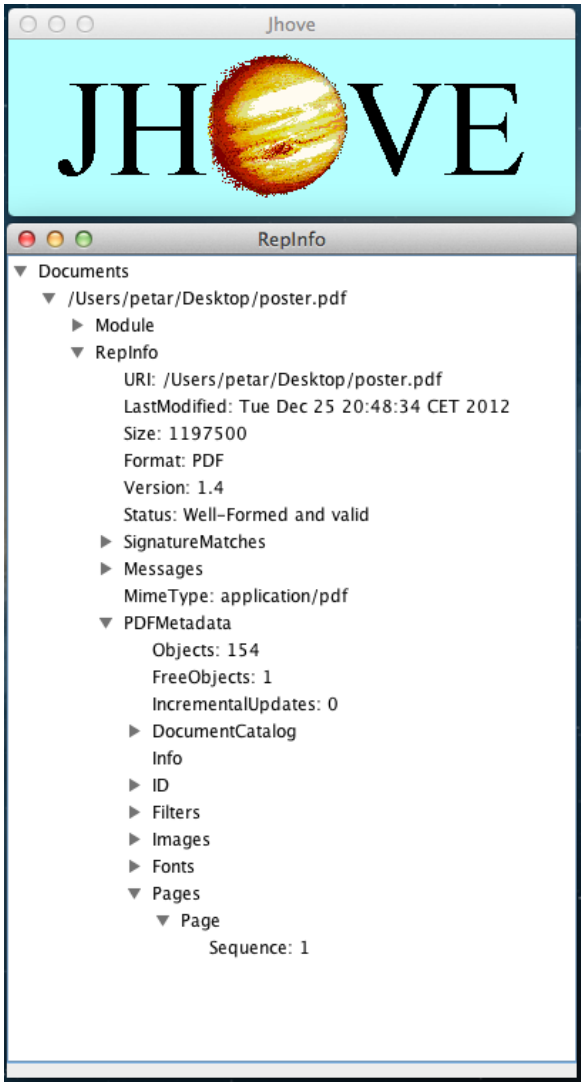

Figure 2.4: A screenshot of JHove showing part of the output for an example PDF file.

```
<?xml version="1.0" encoding="UTF-8"?>
<fits xmlns="http://hul.harvard.edu/ois/xml/ns/fits/fits_output" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://hul.harvard.edu/ois/xml/xsd/fits/fits_output.xsd" version="0.6.0" timestamp="12/14/11 12:39 PM">
 <identification>
   <identity format="Portable Document Format" mimetype="application/pdf" toolname="FITS" toolversion="0.6.0">
     <tool toolname="Jhove" toolversion="1.5" />
     <tool toolname="file utility" toolversion="5.03" />
     <tool toolname="Exiftool" toolversion="7.74" />
     <tool toolname="Droid" toolversion="3.0" />
     <tool toolname="NLNZ Metadata Extractor" toolversion="3.4GA" />
     <tool toolname="ffident" toolversion="0.2" />
     <version toolname="Jhove" toolversion="1.5">1.5</version>
     <externalIdentifier toolname="Droid" toolversion="3.0" type="puid">fmt/19</externalIdentifier>
  </identity>
 </identification>
 <fileinfo>
   <size toolname="Jhove" toolversion="1.5">880359</size>
   <creatingApplicationName toolname="Jhove" toolversion="1.5">QuarkXPress(tm) 6.0/QuarkXPress(tm) 6.0</creatingApplicationName>
   <lastmodified toolname="Exiftool" toolversion="7.74" status="SINGLE_RESULT">2011:12:14 12:37:56+01:00</lastmodified>
   <created toolname="Exiftool" toolversion="7.74" status="SINGLE_RESULT">2004:02:03 11:20:36Z</created>
   <filepath toolname="OIS File Information" toolversion="0.1" status="SINGLE_RESULT">/home/xxx/fitstemp/235/235062.pdf</filepath>
   <filename toolname="OIS File Information" toolversion="0.1" status="SINGLE_RESULT">235062.pdf</filename>
   <md5checksum toolname="OIS File Information" toolversion="0.1" status="SINGLE_RESULT">f284c2925668f9189726b41e051e710a</md5checksum>
   <fslastmodified toolname="OIS File Information" toolversion="0.1" status="SINGLE_RESULT">1323862676000</fslastmodified>
 </fileinfo>
 <filestatus>
   <well-formed toolname="Jhove" toolversion="1.5" status="SINGLE_RESULT">true</well-formed>
   <valid toolname="Jhove" toolversion="1.5" status="SINGLE_RESULT">true</valid>
 </filestatus>
 <metadata>
   <document>
     <title toolname="Jhove" toolversion="1.5">7393.14.02 DAWN_Bprofiles</title>
     <language toolname="Jhove" toolversion="1.5">en-US</language>
     <pageCount toolname="Jhove" toolversion="1.5">64</pageCount>
     <isTagged toolname="Jhove" toolversion="1.5" status="CONFLICT">no</isTagged>
     <isTagged toolname="NLNZ Metadata Extractor" toolversion="3.4GA" status="CONFLICT">yes</isTagged>
     <hasOutline toolname="Jhove" toolversion="1.5">no</hasOutline>
     <hasAnnotations toolname="Jhove" toolversion="1.5" status="SINGLE_RESULT">yes</hasAnnotations>
     <isRightsManaged toolname="Exiftool" toolversion="7.74" status="SINGLE_RESULT">no</isRightsManaged>
     <isProtected toolname="Exiftool" toolversion="7.74">no</isProtected>
     <hasForms toolname="NLNZ Metadata Extractor" toolversion="3.4GA" status="SINGLE_RESULT">yes</hasForms>
   </document>
 </metadata>
</fits>
```

Listing 2.1: An example FITS output file

Although this list is not complete, and there are many other tools that are able to extract meta data as well, it shows that the current state of the art is able to provide enough meta data that could be used as input for various preservation activities. The tools have their downsides in terms of format coverage and/or performance, but still provide very valuable information. Currently there are only a few tools however, that are able to analyse collections. PRONOM ROAR, for example, is able to create a format profile within a repository interface with the help of DROID. Various repositories provide basic information as the formats and size of objects, however no further stratification is possible, although the characterisation data is present. PLATO provides excellent decision making support by utilising different means, but still handles the content profiling step as a high-level non-automated task, which increases the risk of bias during experimentation and analysis. Nonetheless, the means for in-depth analysis on larger-scale are present and seem to be feasible, although there are various issues that still have to be overcome.

2.4 Quality Assurance of Measures

One of the biggest downsides of all identification and characterisation tools is the lack of quality assurance processes. Often there is no way to verify if an extracted measurement value is really representing the truth. This is a huge problem, as it is hard to make assumptions about correctness without having a ground truth benchmark in the first place. Unfortunately, for many different kinds of objects such ground truth data is often unknown and has to be manually harvested from the objects themselves. Due to the complexity of establishing such ground truth data, current approaches often lead to non-reusable data. Becker et al. discuss some of these issues in [4].

Some tools such as FITS try to tackle this problem on a very basic level by encoding a confidence level in the form of an enumeration:

- *OK* - all tools have provided the same measurement value for a given characteristic.
- *SINGLE_RESULT* - one tool provided a measurement value for the given characteristic.
- *PARTIAL* - a subset of the tools have provided the same measurement values for a characteristic.
- *CONFLICT* - two or more tools provide different measurement values for a given characteristic

This strategy is not perfect, but it provides the user with warnings about potential threats. A big problem here is the consolidation. Often tools provide the same measurement for a specific property but the output format is slightly different. This makes it hard for an automatic consolidator to decide if the values are equal or not. Thus the problem of quality assurance depends on external information provided by other processes or even by manual verification. Clearly, this is a hard, tedious and long running process and thus it is often neglected. Nonetheless, it is an essential precondition in order to assure correct input data for the analysis.

As the quality of a content profile is highly dependent on the quality of measurements, these problems have serious impact on the end result of the process. As soon as there are better tools

and approaches for establishing such ground truth data from benchmarks and the quality of the provided measurements is better, the content profile quality will also increase. On the other hand, increasing quality of aggregation and analysis processes will not only contribute to better understanding of the data, but will also create incentives for better characterisation processes that produce high quality meta data.

Nonetheless, the solution presented in this thesis tries to abstract from the problem of quality assurance as it will be addressed in future work and other projects.

2.5 Scenario

The following figures provide an example scenario of the interactions between different components in a simplified swim lane diagram. This example helps us illustrate the responsibilities of each component in the preservation environment and also the usual sequence of events that will occur in such a use case. Afterwards we provide some observations about the current state of the art according to this example.

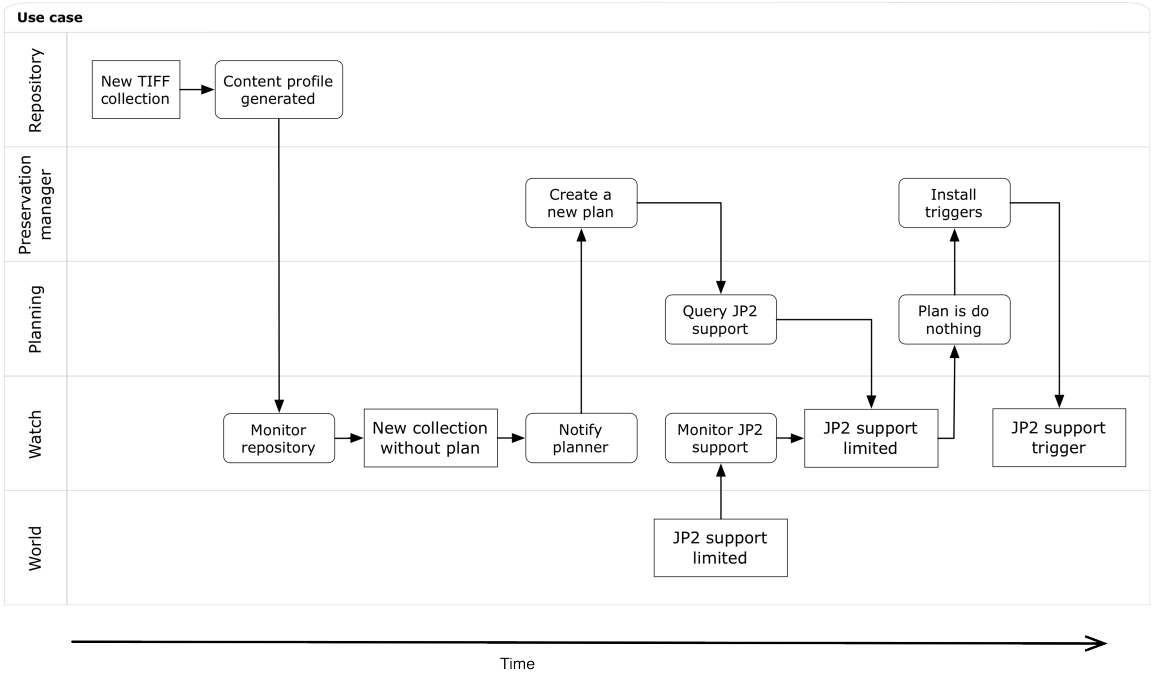

Figure 2.5: The first step of an example preservation planning use case and the interaction between the different components through time.

In this example an image collection that is TIFF 5.0 formatted is examined and a potential migration to the JP2 format is considered during planning. The different components that take part in this scenario are the repository, which manages the collections and provides a profile, the preservation manager, who is a human actor that has the expertise to take valid preservation

decisions, a planning tool, which supports the preservation manager in his activities, a watch component, which is a monitor of internal and external activities and the world, which represents all preservation related information internal and external to the planning environment.

Figure 2.5 shows that the collection is ingested within the repository and a content profile is created.

Afterwards the monitoring component catches the new ingest event and collects the aggregated content profile. The monitor detects that this new collection does not have an associated plan and notifies the responsible planner.

The preservation expert creates a new plan and checks the requirements and policies of the organisation. Assume that one of the objectives is to use only formats that have widespread tool support. In this case the JP2 format still has a limited support, which results in a plan recommendation to keep the status quo. The preservation manager then installs some triggers to the monitoring component that starts observing its sources for changes in the tool support of the JP2 format.

The monitor continues its work and periodically checks for changes in the JP2 format tool support.

Assume for the sake of the example that at some future point in time (Figure 2.6) the JP2 format becomes more popular and thus the number of tools that can produce and render it grows. The monitor component will catch this change and will notify the planner that there is a plan about a certain TIFF collection that needs to be re-evaluated.

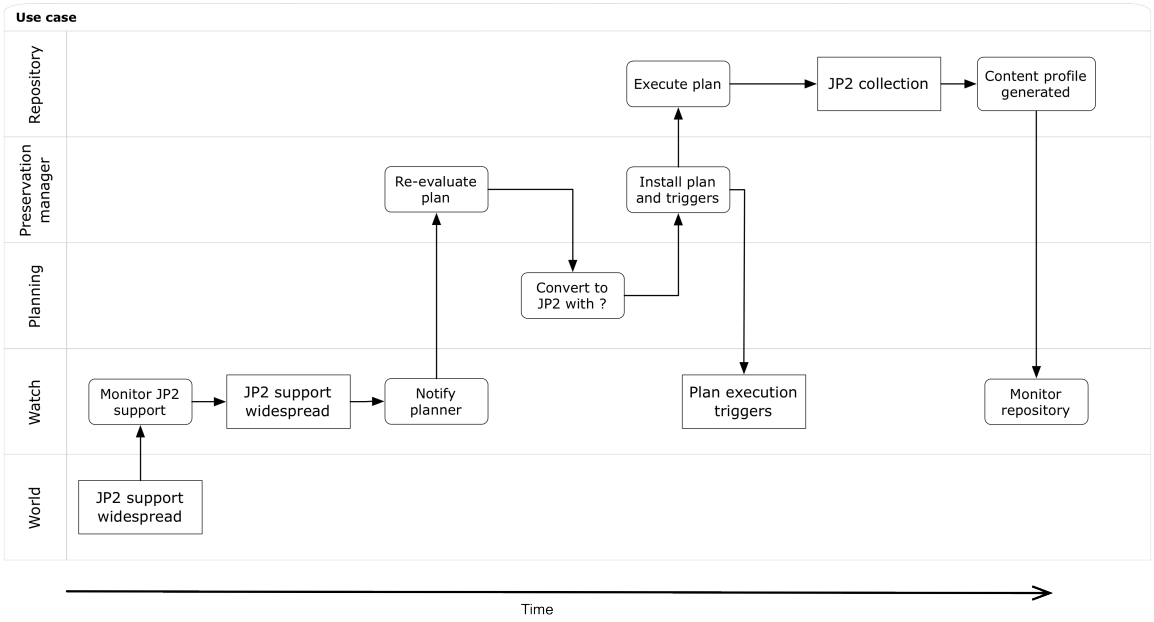

Figure 2.6: The second step of an example preservation planning use case and the interaction between the different components through time.

The preservation manager will conduct experiments and will create a preservation action

plan that will be deployed within the repository. Before execution starts, additional triggers are installed within the monitoring component that keep track of the execution of the migration within the repository.

Once the repository executes the migration from TIFF 5.0 to JP2, the new collection is profiled again and the monitor obtains its new information and continues its work. Thus the whole lifecycle is completed.

This realistic example is summarised in the following simple list of more general steps, which usually apply to many real-world scenarios in the following order:

1. Organisational policies about the management of the content are created and curated.
2. A monitoring component is used to observe the policies and operations over the content within a repository for violations.
3. As part of the repository ingest, an identification and deep characterisation process extracts valuable meta data and stores it within the repository.
4. A content profile is generated and exported for other tools.
5. A monitoring component identifies that a certain subset of objects violates the policies and notifies a planning expert for the potential threat.
6. A planner uses the content profile, a profiler to analyse and stratify the content into smaller homogeneous partitions as well as identify representative sample objects for the partitions.
7. A planning tool is used to validate the threat and to create an action plan that is able to cope with the policy violations.
8. The action plan is submitted to a repository, which knows how to execute the described preservation action.
9. The monitoring component observes the operations of the repository and notifies the interested parties of important events, such as throughput, failures and task executions.
10. The violation is taken care of and the monitor component continues its work.

The state of the art provides some of these components and actors in this high-level workflow. Any organisation actively doing digital preservation has its own set of policies of how to handle different types of content. The problem here is that often such policies are just high-level descriptive statements, which are not structured in any specific form and thus are not machine readable. This makes it almost impossible to use in automated fashion.

Also there are numerous repositories that can manage digital objects and keep them accessible over time. They have the capabilities to extract meta data from them.

The planning tool PLATO provides the needed facilities to create an action plan and support the decision making of a preservation expert.

However, a couple of components are missing. A preservation monitor that scans relevant properties of the world and evaluates their values for changes, eventually notifying users or

other agents of interesting events. Such a monitor is currently developed within the SCAPE Project [1, 11].

Another crucial part that is missing is a method able to profile a massive content set and stratify it into smaller homogeneous and manageable partitions. It can run within the repository or even external to it, but has to provide external interfaces for integration with planning and monitoring. Such a framework will support analysis and content stratification and thus provide better foundation for planning experiments with reduced bias.

CHAPTER 3

Content Profiling

This chapter gives a theoretical overview of the issue of content profiling for digital preservation. It summarises the goals and requirements of the process. Then a detailed definition of the preservation planning process is given, followed by the relation of content profiling to DP and how it fits in the bigger picture. The chapter defines the content profiling process and its necessary steps as well as gives insight into the importance and theory of representative sets. At the end, the topic of continuous profiling and its benefit to preservation systems and experts is discussed shortly.

3.1 Goals

Content Profiling makes use of the output of low-level technical processes and transforms it into input for the high-level decision making process of preservation planning.

Lower level technical processes such as characterisation play an important role not only to preservation planning but also to DP in general. These processes have to deal with single objects (even though the scale can be very large). As output they provide information to quality assurance activities and workflows as well as preservation planning. The identity and meta data extracted out of every single digital object is the scaffold that enables many other processes and applications to do their work. Consequently, the tools that provide this data influence any subsequent process that relies on this data and its validity.

Preservation planning is a high-level process that acts upon sets of objects. As such, preservation planning deals with large volumes of data but is responsible that every single object complies to the requirements of a preservation expert after a preservation action is conducted. The usual size of real world-scenario collections does not allow the test and inspection of every single object before and after each evaluated preservation action. Thus, a bigger picture of the content at hand plays an immense role in the choosing of representative samples and implicitly on the decision made by a preservation expert.

In order to ensure that these two different important parts of Digital Preservation fit together and provide a valid and effective outcome, an adaptor is needed. It has to transform the fine granular output of one process into an aggregated higher level input to the other.

Identification and characterisation are technical processes that can be conducted in automated fashion. Preservation Planning, on the other hand, is a process that can be automated only to a certain extent and will always rely on a human decision. Nonetheless, the degree of automation can be highly improved as the current state of the art is.

A huge problem lies in the fact, that meta data extracting tools often provide data, that is not necessarily valid. The sparsity of the data presents another difficult task when evaluating content during preservation analysis and makes it even more difficult to grasp the peculiarities of the content. The volume of the content and the meta data is high enough to present scalability challenges. Thus the processing is sometimes considered infeasible.

All these issues outline the goals of the content profiling process, which are summarised as follows:

G1 Enable automatic and scalable aggregation of sparse meta data provided by identification and characterisation tools.

G2 Create and expose a well defined, machine-readable footprint of the content at hand, for other actors to use. This has to summarise the following information.

- **G2.1** An identifier of the content or collection.
- **G2.2** The characteristics of the content, such as mime types, formats, size, but also any other that is of interest to preservation planning.
- **G2.3** A set of sample records.
- **G2.4** The scope of the profile (e.g. the object identifiers of the collection, within a repository interface) or some other means to distinguish the objects that conform to this profile.

G3 Enable planning experts to analyse the content. This includes:

- **G3.1** obtaining an overview of the content types and formats, of the collection, but is not restricted only to these.
- **G3.2** generating statistical reports about the size of the content.
- **G3.3** filtering the content into homogeneous sub sets, based on multiple characteristics.

G4 Select representative subsets based on different approaches, that make sense in different planning use cases.

G5 Browse the raw meta data of objects.

G6 Export the raw (sparse) meta data in a common format that can be processed by other analysis software, which will enable preservation experts to conduct even more in-depth analysis.

3.2 Profiling

This section describes content profiling and discusses its prerequisites. Afterwards it proposes an approach of creating a profile in an automatic fashion. Subsequently the enhancement of preservation planning through automation support during analysis of content in real world DP scenarios and integration with other DP information systems is discussed.

Prerequisites

In order to generate a valid content profile, some prerequisites have to be met. For one, the characterisation has to be provided, but also its data quality in terms of validity, normalisation, etc. has to be guaranteed.

Characterisation Data

Clearly, to profile a set of digital objects, the meta data of the set of objects are needed. There are arguments whether or not the characterisation process should be part of profiling. We support the opinion that characterisation should be done before and its output serves as input to a profiler. There are two main reasons for this. One, in real world scenarios content is usually stored in special archives (digital repositories), which extract the meta data out of the digital objects upon ingest in these systems. Thus, it does not make sense to access the original objects again and run a potentially time-consuming process over them again. The second reason is that profiling is an analysis step and should not make use of one characterisation tool, but should try to be agnostic to the meta data format. After all meta data are just key-value pairs. It is much better to support different formats and transform them to an internal model instead of restricting the whole process to one format.

Data & Normalisation

In order to achieve the goals outlined above, the meta data provided by the characterisation tools has to be normalised and should fit into a unified model no matter its origin.

Consider the following example. Two characterisation tools provide meta data for document formats and measure different characteristics for these documents but one - the number of embedded tables in the document. Only, the first tool reports this characteristic under the name *'tableCount'* and the second *'nrTables'*. Semantically both measures provide the same information. Thus it will be very valuable to a planner to know if both tools provided the same result or not. The coverage of this property within the characterised set will be potentially higher, since characterisation tools do not often manage to extract every property out of every digital object.

This example reveals two problems. Firstly, if the data is not normalised at all, the sparsity will be even higher and this will compromise the analysis and the gained knowledge about the collection. The second problem is the difficulty of normalisation itself. In the given example, it is fairly easy to assume that the semantics of the given property are the same. However, this is not true for all characteristics of all properties. Correct normalisation requires the domain knowledge of DP experts, the developers of the characterisation tool and potentially other experts. If it is done wrong, the whole result of a profile can be compromised.

Because of these problems, it is very important that characterisation tools provide valid and well-documented data. On the other hand, profiling tools should allow the normalisation of such data if the characteristics are well defined and it is clear that the semantics of two or more properties are the same.

For this purpose, a simple but flexible domain model has to be created that allows properties, their measurements, provenance information and more to be stored at one place. The data structure has to enable efficient aggregation and querying of the data. Important aspects that have to be taken into consideration during design are the sparsity of the meta data, the different data types of the measurements. As discussed, the validity of measurements is often unreliable, so it is important to keep the provenance information of the measurements. If the profiler is ought to keep track of continuous data, then the time of measurement should also be taken into account. Another important issue is the identifier of each object, which has to be able to point to the original in the repository system.

The process

Content profiling consists of three main parts; data harvesting, data aggregation and analysis. These are illustrated in Figure 3.1 and are discussed in the following.

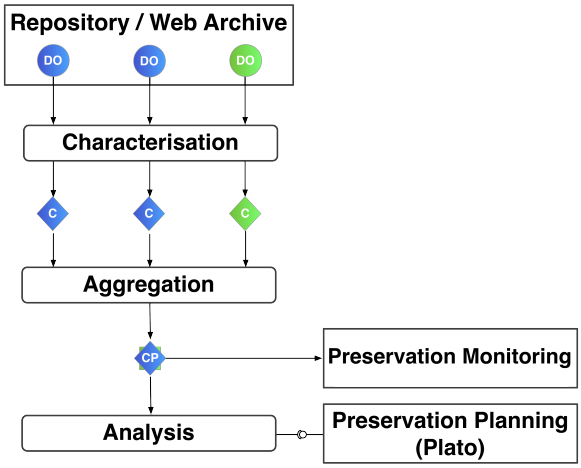

Figure 3.1: The three steps of content profiling

The first step is responsible for characterising, collecting and processing the digital objects (DO) in a DP system and adapting it to the internal model of the profile framework. This includes normalisation of the different properties and characteristics, removal of unnecessary data and more. Since DP scenarios usually make use of digital repositories, it should be possible to harvest the meta data directly from the repository. This can present a potential issue, due to the differences in the interfaces of repositories. Nonetheless, it has to be considered within the design of the system. Of course the approach could include characterisation as part of the process, which would result in a collection of meta data files. In such a case, the disadvantages

discussed above, should be considered. After extracting and parsing the characterisation data out of the digital objects, special post-processing steps can be applied to each meta data object in order to refine the gathered data. For example, if the characterisation process does not provide normalised data, special actions can be undertaken to deal with this issue. Also, if there are conflicted values of the same property provided by different meta data sources, data cleanup actions and rules can be executed. These post-processing steps can have huge impact on the final result and thus should not be executed lightly, but only through special configuration steps, done by preservation experts, that understand the source of the meta data, the operations of the identification and characterisation tools and the implications of such alterations of the data. Thus, such functionality of the content profiler has to be done in a flexible fashion and has to allow special configuration that can alter and turn on and off such behaviour. Once the data fits the internal representation of the profiler, it can be stored in a database per digital object level and thus allow further processing by the profiling process.

The next step of aggregation can be done either after the characterisation data (C) is parsed, normalised and stored or on demand if the underlying data store provides the necessary facilities. It is responsible to present the big picture of the data in a smaller footprint and structured form, so that other programs and tools can integrate with it. This content profile (CP) should provide enough flexibility to understand the data but also be aggregated as much as possible. Here different queries and aggregations can be executed, stored or cached in order to enable analysis or export of the data. An important issue that has to be considered is how the content profile data is going to be kept in sync during longer timeframes and after content changes.

The last step is a high-level service on top of the framework, that has to be conducted mostly manually by the preservation expert. This means that it will rely on specific decisions and will need the input of a user. Nonetheless, it should be automated as much as possible and shall support the user in her decisions. This includes querying the data and gaining knowledge about the content by drilling it down based on different characteristics, but also the export of the whole raw data or a subset of it in a representation that allows further analysis but provides comprehensible overview. Due to the nature of the data (a set of key-value pairs per object), a sparse matrix seems to be a good fit for this.

Limitations & Pitfalls

There are a number of problems and potential pitfalls that have to be considered when designing and implementing the proposed framework.

A very important issue is the removal of objects. It is, unusual that objects are removed from archives, but it is not impossible (e.g. after migration). Once an object (or a set of objects) is removed, the profile may become invalid and needs to be regenerated. If the source system (repository, file system, etc) does not provide information of which objects were deleted, the whole process should be repeated from the beginning. This could be time and resource consuming.

The same holds if new meta data is acquired and merged within the current profile. Here normalisation has to be kept intact and also any aggregation that might be influenced should be recalculated.

Clearly, the quality of a content profile is highly dependant on the quality of meta data, which makes it very important for preservation experts to know and understand the identification and characterisation tools they use.

If data normalisation is done via the framework and is not part of the meta data input, then very specific expertise will be required to do this task efficiently.

Output

The profiler has to present the aggregated data in a form that is usable to planning experts, but also allow the export of the data into other formats for further processing. Furthermore, the preservation process will be enhanced only if the data can be obtained in a machine-readable format, so that other tools can interact with it.

There are numerous ways of representing such a profile. However, from the observations of the previous chapter there are number of goals that it has to fulfil. It has to follow a well-defined schema; the profile has to be in a structured, machine-readable form, as it will act as input to other information systems in the DP landscape.

Clearly, it has to aggregate the raw data, so that it has a small enough footprint, but still provide enough insight into the content at hand. The profile should be able to separate the content based on different characteristics. Here, it is important that a user or a user application is able to go one step back and examine the filtering criteria. Finally, the profile should include a small set of representative objects, with their full characteristics and a way to identify all objects in a set or collection that fit into the specified filter. The latter could be done, either via some kind of query or just by a list of objects outlines the identifiers of all selected objects.

Having such a content profile is the foundation for integration with other DP software systems, such as preservation watch and monitoring systems, simulation environments and planning components. For this integration some high-level features based on the exported profile and representative sets can be done.

Listing 3.1 proposes a possible content profile schema for a XML representation of a profile. A profile consists of a sequence of partitions. Each partition defines a subset of a whole collection, in other words it defines a scope of the profile. The partition has four main sections capturing aspects such as the filter of a partition or the scope, the property aggregations, samples objects, and optionally object identifiers. The filter is currently left to be any sequence of valid XML. This allows defining a filter that is application-specific and can be used in order to select the exact same partition within the profile tool that generated the profile. Although it is not a perfect solution, it had to be chosen because there is no common unified repository interface that allows the selection of data based on property value pairs. Enhancing this part is an interesting future work, especially if repositories implement a standard query mechanism. This will allow a plan to read the query out of the profile and automatically be applied over all objects in the repository that have the same scope. One option for such an optimisation may be the use of a specification called Search and Retrieve by URL - SRU[1]. It was developed by the Library of Congress for the purpose of encoding special queries within a URL and has common libraries that can be used by repositories or other data managing software. The properties section sum-

[1] http://www.loc.gov/standards/sru/

marises important properties for the profile and aggregates them. The profile also allows that the properties include a distribution of their values. The samples section includes different sample elements with all known property-value pairs and the tool that reported them. The last section is optional and presents a list of element identifiers in the partition.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://ifs.tuwien.ac.at/dp/c3po"
       targetNamespace="http://ifs.tuwien.ac.at/dp/c3po" elementFormDefault="qualified" version="1.0"
       >
3    <xs:element name="profile">
4      <xs:complexType>
5        <xs:sequence>
6          <xs:element ref="partition"/>
7        </xs:sequence>
8        <xs:attribute name="collection" use="required" type="xs:string"/>
9        <xs:attribute name="count" use="required" type="xs:integer"/>
10       <xs:attribute name="date" use="required"/>
11     </xs:complexType>
12   </xs:element>
13   <xs:element name="partition">
14     <xs:complexType>
15       <xs:sequence>
16         <xs:element ref="filter"/>
17         <xs:element ref="properties"/>
18         <xs:element ref="samples"/>
19         <xs:element ref="elements"/>
20       </xs:sequence>
21       <xs:attribute name="count" use="required" type="xs:integer"/>
22     </xs:complexType>
23   </xs:element>
24   <xs:element name="filter">
25     <xs:complexType>
26       <xs:sequence>
27         <xs:any processContents="skip" minOccurs="0" maxOccurs="unbounded" />
28       </xs:sequence>
29       <xs:attribute name="id" type="xs:string" />
30     </xs:complexType>
31   </xs:element>
32   <xs:element name="properties">
33     <xs:complexType>
34       <xs:sequence>
35         <xs:element maxOccurs="unbounded" ref="property"/>
36       </xs:sequence>
37     </xs:complexType>
38   </xs:element>
39   <xs:element name="property">
40     <xs:complexType>
41       <xs:sequence>
42         <xs:element minOccurs="0" maxOccurs="unbounded" ref="item"/>
43       </xs:sequence>
44       <xs:attribute name="avg" type="xs:decimal"/>
45       <xs:attribute name="count" use="required" type="xs:integer"/>
46       <xs:attribute name="id" use="required" type="xs:string"/>
47       <xs:attribute name="max" type="xs:double"/>
48       <xs:attribute name="min" type="xs:double"/>
49       <xs:attribute name="sd" type="xs:double"/>
50       <xs:attribute name="sum" type="xs:double"/>
51       <xs:attribute name="type" use="required" type="xs:string"/>
52       <xs:attribute name="var" type="xs:double"/>
53     </xs:complexType>
54   </xs:element>
```

```
55    <xs:element name="item">
56      <xs:complexType>
57        <xs:attribute name="count" type="xs:integer"/>
58        <xs:attribute name="id"/>
59        <xs:attribute name="value" use="required" type="xs:string"/>
60      </xs:complexType>
61    </xs:element>
62    <xs:element name="samples">
63      <xs:complexType>
64        <xs:sequence>
65          <xs:element maxOccurs="unbounded" ref="sample" />
66        </xs:sequence>
67        <xs:attribute name="type" type="xs:string" />
68      </xs:complexType>
69    </xs:element>
70    <xs:element name="sample">
71      <xs:complexType>
72        <xs:sequence>
73          <xs:element minOccurs="1" maxOccurs="unbounded" ref="record" />
74        </xs:sequence>
75        <xs:attribute name="uid" type="xs:string" />
76      </xs:complexType>
77    </xs:element>
78    <xs:element name="record">
79      <xs:complexType>
80        <xs:attribute name="name" type="xs:string" />
81        <xs:attribute name="value" type="xs:string" />
82        <xs:attribute name="tool" type="xs:string" />
83      </xs:complexType>
84    </xs:element>
85    <xs:element name="elements">
86      <xs:complexType>
87        <xs:sequence>
88          <xs:element maxOccurs="unbounded" ref="element"/>
89        </xs:sequence>
90      </xs:complexType>
91    </xs:element>
92    <xs:element name="element">
93      <xs:complexType>
94        <xs:attribute name="uid" use="required"/>
95      </xs:complexType>
96    </xs:element>
97  </xs:schema>
```

Listing 3.1: XML schema for a machine-readable content profile

3.3 Continuos Profiling

A content profiler tool can provide benefit to numerous systems and users in a Digital Preservation environment. On the one hand, it can act as direct input to Preservation Planning and can spare a planning expert a lot of time defining and outlining a plan. On the other hand, it supplies the needed foundation to do continuous profiling. Continuous profiling refers to two high-level applications of content profiling when integrating with other systems. The one is profiling data based on its creation or harvesting date. The other one is the continuous monitoring of a profile of a collection and all its changes through time. Here we give some overview of both ideas and summarise how these can be achieved and what benefits they will provide to planning experts.

Monitoring

As discussed in the previous chapter, monitoring is an external phase of the preservation planning environment that provides feedback and can trigger re-evaluation of a plan. If a content profile generated from the content of a repository is observed continuously by such a monitoring system, then certain triggers can be raised when conditions are met or constraints violated. For example, an organisation that may have a policy that defines which objects have to be preserved and one that defines how many objects of a certain type can exist without a valid plan associated to them. Now consider that each day a great deal of new objects is ingested into the repository of that particular organisation. By the end of a certain time period, a content profile is regenerated and a monitoring system obtains and re-evaluates this result against the organisational policies. It will be fairly easy to detect a potential mismatch and to notify the involved stakeholders and actors in order to resolve it provided all of these pieces of the puzzle fit together.

Another benefit from the combination of such content profile and monitoring system can be a global content profile. If enough organisations such as Web Archives, Libraries, etc. are willing to share the profiles (or parts of them) they have with such a central monitor, the result will most probably be a representative overview of the global state of many preservation related properties of the world that are relevant for DP [1]. Valuable information, such as the number of formats used within a certain domain, the preferred type of files within a user community, the emergence of new file formats, the decline of old file formats and much more can be detected. Provided that enough content-holding institutions are ready to contribute, the mutual benefit could be something much more valuable.

The integration between a content profiling tool and such a monitoring system presents some technical issues and questions that have to be answered, as content holding institutions make use of different repository and systems. Nonetheless, if there is a profiling tool running within such a system or near the data and conforms to a standardised machine-readable output, then such integration is feasible. A design proposition of such a novel preservation watch system is discussed in [11, 12].

Simulation & Trend Analysis

Content profiling can be integrated with simulation software that tries to analyse the trends of different characteristics of the content over time such as future development of format profiles,

detection of format obsolescence and emerging formats, size fluctuations, or any other preservation related characteristic.

Weihs et.al demonstrate a simulation prototype [46] that can simulate the evolution of a repository over time. Different models and configurations of the content, as well as the potential results of preservation actions over this content are used as input. Integration between such a tool and a real world content profile is an approach to examine and validate how simulation influences digital preservation decisions, but also provide new requirements and goals for content profiling as well.

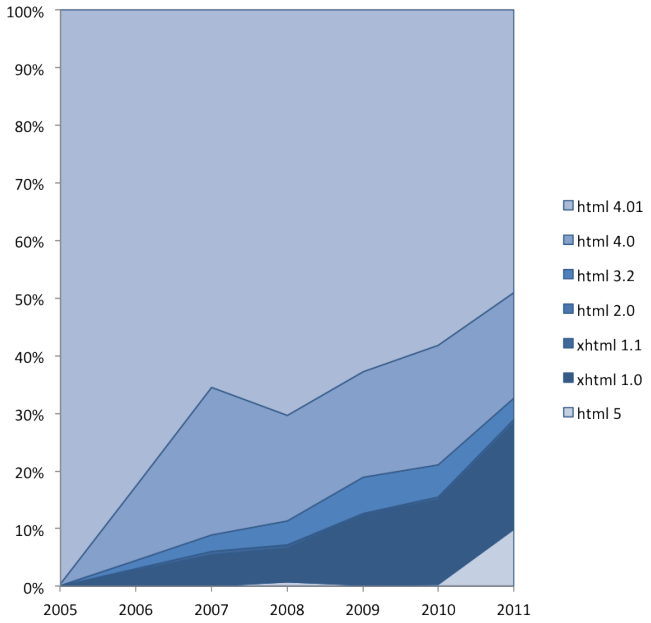

Figure 3.2: Usage trends in html versions over time in a Danish web archive

Other interesting aspects to simulate include the format profile of a repository over time. There are different theories regarding the lifetime of formats and a debate pro and contra for following a direct migration strategy or whether or not this should be left to the so called network effects of data sharing, which should prevent the damages of obsolescence [39]. Jackson gives a good summary of this debate in [23] and asks the question: ”Where is the evidence?” In order to find out, experiments of format usage over large digital corpora of the British Web Archive are conducted. The results suggest that there are indeed formats that are present in the archive longer than five years, however there were a number of formats fading from use and these should be studied closely. These (unstructured) profiles could act as input to simulation environments, where based on the past trends new hypothesis for the future development are tested and used for planning. Another interesting aspect is the differentiation between use of formats in production and in access. There are certain formats where production declines rapidly and still there is a great deal of rendering software that enables the access of the existing content. The question

remains, whether a format is obsolete when evidence for its production stops or when it cannot be accessed and rendered correctly or a combination of both.

One example of such trend results is presented in Figure 3.2. The different HTML versions of about 1.5 million web archive digital objects from the Danish web archive are presented through the years. The data for this graph was processed with the help of the backend of a content profile tool following the suggested approach in this chapter. It was produced for another thesis conducted at the time of writing at the University of Technology in Vienna that researches different statistical trend analysis approaches for preservation. Thus, we believe that such content profiling output data can be of very high value to DP activities if it can be structured and processed by other software.

3.4 Representative Sets

As discussed in Section 2.1, finding a small set of representative sample objects is one of the most important steps during the first step of the planning process, as it forms the foundation for the chosen course of action.

The goal is to choose a small set of objects out of a much larger set. This small set should ideally capture the essence of the large set, which means covering as much of the different measurements of a set of characteristics [5].

There are a couple of questions that a preservation planner has to answer when selecting this set. Firstly, how many samples are enough? And secondly, what characteristics are important for the coverage?

Unfortunately, both of these questions do not have specific answers and are highly dependent on the given preservation use case. Nonetheless, these questions outline some requirements for the planner to consider during selection and planning. The sample set has to have as few as possible entries due to the resource consumption of the experiments, but as many as needed to guarantee the validity of the experiments. As far as the characteristics are concerned, the planner has to select such characteristics that are important for the use case, that are measurable and could cover as much as possible in the set. Consider a collection that is homogeneous in terms of the format. Choosing your samples based on the format would not make much sense, as the chosen samples would still be random due to the format homogeneity.

Choosing such samples is somehow related to the more known problem of clustering. Clustering tries to organise a given set of patterns based on similarity, usually by representing them in vectors or points in multidimensional spaces and their distances [24]. The problem is that we do not only search for similar objects based on different characteristics, but we want to select enough objects of every distinct cluster, so that we have the confidence that enough objects were selected to cover all peculiarities and differences between them, that could be the cause of a failure. To some extent the first step of filtering through the content and finding a homogeneous set is the clustering, which results in one cluster of the whole content. The representative sample set is a much smaller subset in this cluster that has to cover as much as possible of the differences between the objects inside of the homogeneous set. Clearly, if the first step is omitted, finding a representative set should still work. However, it will be more complex to achieve a good coverage, as the differences will be much more as well as the data sparsity larger.

CHAPTER 4

C3PO - A Profiling Tool

This chapter summarises the goals of the content profiling approach discussed so far and gives a detailed overview of the architecture for a prototype implementation of it. First a high-level design is discussed and then detailed explanation of the chosen technology, trade-offs, implementation details and enhancements made is provided. Afterwards some algorithms for representative sample selection that were implemented are presented and discussed. At the end a short summary and overview of future interesting work which will enhance the tool and help preservation experts even more in their endeavours is given.

4.1 C3PO in Perspective

As part of this thesis, a software prototype is developed that aims to tackle the issues, problems and gaps presented in the previous chapters. The aim is to create a tool that is able to automatically generate a content profile of large collections (consisting of hundreds of thousands, or even millions of objects). This means a framework that provides enough scalability to be feasibly applied on the metadata of collections of multi-terabyte ranges and beyond.

This profile includes a descriptive statistical representation of the content in the collection, meaning that it contains very specific data such as count of objects, overall size of objects, etc. Furthermore, it provides visualisations of different aspects of the content, such as the file formats, mime types and many other characteristics and combinations thereof that are of interest to the user of the application.

As presented in section 3.1 on page 31, the tool that implements the presented approach should fulfil the following requirements.

R1 The user should be able to aggregate large sets of meta data.

- **R1.1** This should happen with minimal effort spent for setup and configuration and happen in an automated fashion.

R1.2 The process should finish in a time frame short enough, so that the information in the generated profile is still up to date.

R1.3 The time needed for the process to finish should scale linearly to the size and volume of the collection.

R2 The presented approach should scale up to a million objects and should be ready to pass this threshold by making use of a distributed infrastructure and architecture.

R3 Clients of the application (users, or other systems) should be able to obtain a machine-readable description of the profile, that contains an overview of the content at hand.

R3.1 This profile shall include the identifiers of a small set of objects that are representative to the whole collection, as described in section 3.4 on page 41.

R3.2 The profile shall include an identifier of the collection as well as the object identifiers it refers to.

R4 Planning experts should have the ability to visualise different aspects of the content.

R5 Planning experts should have the ability to filter the content based on chosen characteristics.

R6 Planning experts should have the ability to present the raw meta data or a subset of it in a sparse matrix view, where each row is a digital object, each column is a property, and each cell has the value of the corresponding property for the corresponding element.

The prototype implementation of the profiling approach is called '**C**lever, **C**rafty **C**ontent **P**rofiling of **O**bjects' or *C3PO* for short and is discussed in detail in the following subsections.

4.2 Architecture

In this section an overview of the architecture of C3PO is presented. Afterwards, detailed information about the design and implementation is given as well as a discussion of the decisions made is done. The implementation was carried out in two iterations, which are presented after the overview.

High Level Overview

C3PO is separated into different modules and provides a relatively simple workflow that follows the three steps of content profiling as presented in [33] and discussed in section 3.2. In the first part, it gathers raw meta data and parses it in order to normalise it into a simple internal data model. In the next step, the data is cleaned up, partially aggregated and stored into a database.

All this offers the baseline for the deeper analysis provided by some of the modules of the framework. Figure 4.1 presents the high level architecture and C3PO's modules in a stack diagram that is detailed in the following subsections.

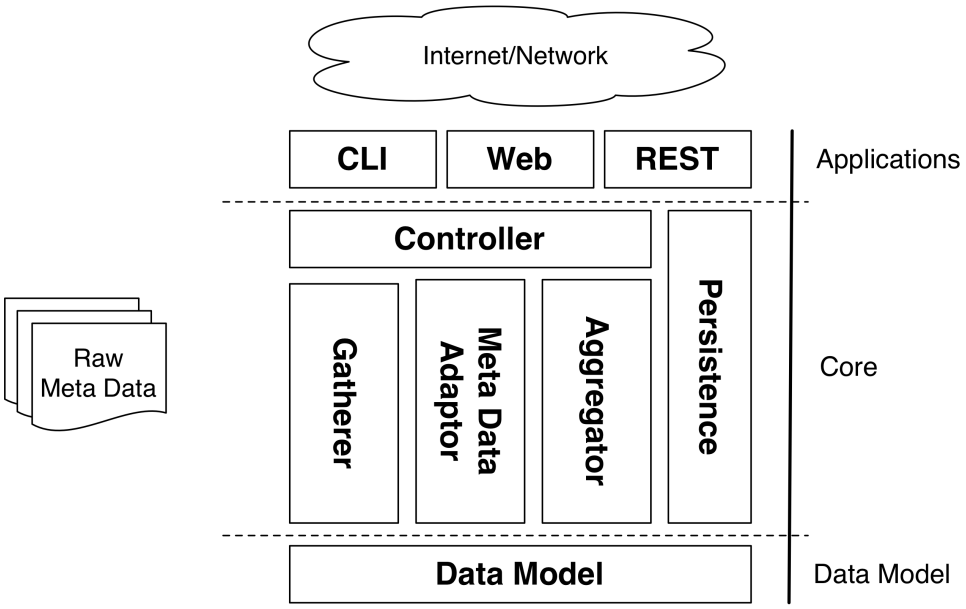

Figure 4.1: High-level architecture of C3PO.

Data Model

At the bottom is the domain model module that sits as the foundation of the architecture. It represents a simple model that captures the important aspects of the content meta data in a generic way, but still provides the ability to do flexible queries over the data. It consists mainly of elements, properties and values. *Elements* encapsulate a digital object and capture important information, such as the identifier of an object within its source (e.g. a digital repository), which is important for later access. *Properties* define characteristics of a given object. These may include information such as size, format, format version, etc. The *Values* capture measures of specific properties that are provided by different tools.

Core

Above the data model is the main part or the core module, which encapsulates the framework that allows the gathering, normalising and aggregation of the data. It not only offers interfaces that allow the extension of the framework, but also provides the glue-ware to create and run the workflow.

The core wraps the middle part of the architecture stack diagram. It is divided into three logical parts: A controller, sub-components and a persistence layer.

The controller is responsible for chaining the sub components and carrying out the whole workflow of the system.

The three sub-components (*Gatherer*, *Meta Data Adaptor* and *Aggregator*) divide the work and encapsulate important steps of the whole workflow. The raw meta data of the content can

be stored in many different places or sources. For example, it can be provided locally, in form of files stored on the local file system, or remotely. The remote source can have many different variations, e.g. there can be a remote SSH server that stores the raw meta data again in file form, or a remote web archive server that stores the meta data in special container files called ARC (archive) or WARC (web archive) file, but there can be also a remote repository, which not only stores the original content but all the meta data for each object in a different way (internal data store, to its local file system, etc.). The latter represents the most likely use case in a real world digital preservation scenario. However all others are possible as well and in fact are easier to use for experimentation purposes.

As the users of C3PO should not be interested in the way of how and where the meta data is stored, the gatherer component offers an interface that abstracts this issue.

Since different sources can use different characterisation tools and different characterisation tool outputs, the Meta Data Adaptor component is responsible for instantiating and assigning a specific implementation of an adaptor that can handle the gathered meta data records.

The last part of the core module is the persistence layer, which abstracts the connection to the external data source, where all raw meta data is stored. It provides interfaces for retrieving the meta data, storing aggregations and analysis results.

Applications

On top of the Core module, there are two user applications. One of them has a command line interface (CLI) and the other a web application user interface. This separation has been done in order to optimise network overhead during initial data processing. The CLI application can be executed near the data (e.g. on the same infrastructure as the repository) and can process and store the data there. This near-data processing allows the reduction of network overhead and the utilisation of resources. The web-application, on the other hand, can be deployed on a different infrastructure and provides interfaces for analysis and representation of the data.

First Iteration

The first iteration of the implementation was done in order to explore the problem of content profiling and find out potential issues early on. It was a horizontal prototype and concentrated on the lower levels of the framework.

As relational databases have stood the test of time and are proven to work in virtually any use case, the natural thing was to try a relational model first. Figure 4.2 shows a simplified version of the key domains of the first data model used. It models the key concepts for generic key value structure in a relational database, which would fit the needs of the collection profiler and leaves out some fields and helper classes.

The *Elements* describe the digital objects. Each element has a number of *Values*, where every value is a measurement for a specific *Property*. Properties are specific characteristics, such as format, size, or number of pages in a document, etc.

There are different typed values for the different data types, such as String, Bool, Numeric, Float and Array, which are not shown in the diagram. Furthermore, each value has a *Value-Source*, which provides provenance information.

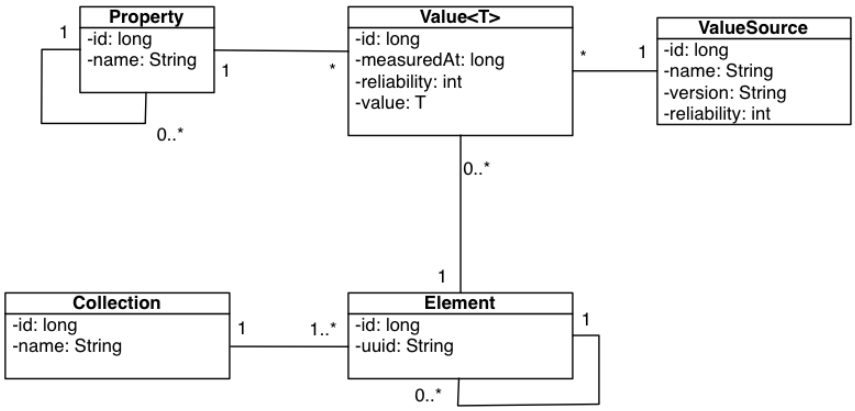

Figure 4.2: The data model used in the first iteration.

Although this model was so minimalistic, it was proven to be incapable of a sufficient performance when querying more specific information (a mixture of more properties and values) due to the generic nature of the Values. This proved to be a big problem in regard to the sparse matrix export requirement (**R6**). This use case provides the user with a great overview of the data and enables her to find important aspects that would otherwise easily evade. Since the data is sparse, it was not feasible to query the described matrix of the data in an efficient way, which made the implementation of this key requirement hard.

The underlying data store was a PostgreSQL database, which is one of the best open relational databases at the time of writing. However, the limitation of the high number of JOINS for the sparse matrix use case is contradictory to the paradigm, since there is a general rule of thumb that more JOINS result in a poor performance. Through data model enhancements and optimisations and query optimisation, it may be possible to use a relational model effectively. However, this was not the focus of the work, so a new approach was chosen.

In the first implementation the *Controller* worked in a single threaded, sequential manner due to simplicity reasons. However, preliminary experiments have shown that some tasks, such as meta data parsing make only partial use of the system resources at hand. For example the CPU utilisation was mostly less than 25% during data parsing and storage. This was an indicator of a possible enhancement that might result in a significant speedup.

Because of the XML representation of the FITS meta data, a parser was written in order to adapt the meta data schema of FITS to the internal data model. As FITS files usually are small in size (up to 5KB), the parser that was implemented utilised a DOM based approach. Early tests over a small set of FITS XML files had feasible performance. However, the first test over larger collections revealed an issue with memory usage. Even though the amount of objects that had to be created by the DOM parser for each node in each XML document was not so

high, the garbage collector of the JVM was unreliable and the overall consumption increased constantly with the XML document count and the collection size. This was in collision with the performance requirement (**R2**).

In the first iteration C3PO's persistence layer was based on the Java Persistence API[1] (JPA 1.0) with Hibernate[2] as the persistence provider. The high-level abstraction was done via a couple of generic data access object (DAO) interfaces, which were implemented by the client application, and client modules.

This design was chosen in order to allow each client application to choose its own implementation of the persistence layer. This was important since C3PO is meant to be deployable in application server containers that support the Java EE[3] technology stack. For this, a special container-managed transactional model would have been needed. On the other hand, content profiling is a data intensive process which can gain from the fact that the tool can execute near the data. That is why also a local transactional model was needed, in which the tool should run locally near the data without any application server. All this would allow the separation of the data gathering and the data analysis parts of the workflow. The fact that the database is external to the tool also means that it can be setup near the data or on a specific storage sever that has enough resources at its hand to handle the load.

Using an ORM framework such as Hibernate is often very useful. However, in this instance a lot of optimisations and tweaks had to be done in order get out the most of the database. While this was certainly possible, it was not the focus of the work to fight the framework. Using JVM profiling applications revealed that a lot of resources are used during storage, because of the high volume of data. The ORM framework was the bottleneck.

In the first iteration only a prototype of the CLI application was implemented that made use of the non-transactional persistence model. It was used to measure performance and to monitor the behaviour of the whole framework. As far as the web application was concerned, only the persistence layer interfaces were implemented in order to validate the design and the separation of concerns.

Hence, the exploratory prototype revealed that this use case does not fit well into a relational paradigm and the use of an ORM framework decreased the performance of the process during storage. What is more, parsing the meta data with a DOM parser did not scale well and consumed large amounts of memory. These insights were the key to the design of the delivered prototype for C3PO.

Second Iteration

This section gives an overview of the implementation changes and enhancements in the second iteration and discusses the benefits and drawbacks of the alternatives.

[1] http://docs.oracle.com/javaee/6/tutorial/doc/bnbpz.html
[2] http://www.hibernate.org/
[3] http://www.oracle.com/technetwork/java/javaee/overview/index.html

Data model

After examination of the data at hand, it seemed that the key value structure was fitting, but through the data base normalisation, performance was compromised (**R2**). Thus, an appropriate choice was to exchange the underlying data source, which made it possible to remove the ORM framework at all. Its overhead was proven to be unnecessary during meta data adaptation and storage. Also, the analysis of the data took an unnecessary long time due to the many JOIN operations. While these could have been avoided by making the data model more specific, this would have compromised the flexibility, which was a key requirement.

For these reasons, the data model was analysed again and different storage possibilities were evaluated. Due to the natural key-value structure of the meta data a key value store was assessed to provide a better solution to the problem. However, usually such solutions are used for caching (EHCache[4], Memchahed[5], etc.) and architecture designers often have problems fitting their data model when such technologies are used for more than their purpose - caching. There are some implementations of data bases that offer most of the flexibility of the relational paradigm, high performance due to their almost key-value paradigm and out of the box horizontal scalability.

MongoDB[6] is a document store that uses BSON[7] (Binary JSON) in order to store data in form of documents. These documents can have any kind of structure and do not have to be normalised in the relational database sense. On top of that, MongoDB provides native facilities for executing Map Reduce [10] jobs on the server, which proved to be very useful for aggregating and filtering the data. As any NoSQL solution, MongoDB does not require normalised relational data. On the contrary, NoSQL solutions give up normalisation in order to enhance the performance of specific use cases. MongoDB was chosen because of several reasons:

- **Natural fit of the data into a key-value schema.** The data model from the first iteration was transformed into three collections (here the term collection is used in the sense of a document store and can be understood as the equivalent concept of a table in a relational database): one for the elements, one for the properties and one for the sources. The values are embedded into each element document and thus each document represents a self-contained object with all its known meta data - values, sources and conflicts. This has one big advantage when querying, as all values are already present without a need of joining the property table. Figure 4.1 gives an overview of the basic structure of element documents in BSON syntax. Note that the syntax is very similar to JSON, but provides data type support (e.g. the ObjectId).

- **Support for native Map Reduce jobs** makes it is easy to aggregate specific values of specific keys in every element document or to execute analysis queries. This enhances performance as most of the computation is done near the data.

- **Query cursor and pagination support.** Every query in Mongo DB returns a cursor over the data instead of the data itself. This makes navigation over the data within an

[4] http://ehcache.org
[5] http://memcached.org
[6] http://www.mongodb.org/
[7] http://bsonspec.org/

application fast and efficient in terms of memory, as one does not have to consider the volume of the queried data. This lazy-loading mechanism comes in handy in numerous use cases of the application.

- **Horizontal scaling and automatic node balancing** are supported out of the box. This is useful when considering the amounts of data that the profiler has to deal with. It will also reduce the configuration and administration overhead if such a system is used in production.

```
{
  _id : ObjectId("4fe1d1c00364689befd96b62"),
  name: "MyObject",
  uid: "/some/unique/identifier/within/the/source",
  collection: "MyCollection",
  metadata: {
    key_one : {
      status : "OK",
      value: "some value",
      sources :  ["..."]
    },
    key_two : {
      status : "OK",
      value: "other value",
      sources:  ["..."]
    }
  }
}
```

Listing 4.1: The structure of a C3PO element represented as a BSON document.

Core

In the second iteration, the *Controller* and the meta data adaptors were implemented again following a master-worker pattern. The Controller uses the *Gatherer* interface to traverse the raw meta data files and spawns an adaptor worker thread for each file that has to be processed. This change resulted in much larger utilisation of CPU resources. Each meta data adaptor runs in a thread and is responsible for parsing meta data files that conform to a specific meta data schema. The C3PO prototype makes use of a single adaptor for the FITS output format as described in Section 2.3.

The parser implementation of the FITS adaptor was also changed due to the previous performance results. This time, a SAX based approach was used. The Apache Commons Digester library[8] provides a special SAX parser that traverses each document only once and does not require the building of a complex DOM tree. This results in a much faster parsing with significantly less memory resources.

[8] http://commons.apache.org/digester/

Applications

The CLI application was modified so that it reflects the changes of the new persistence layer. This allowed for faster and easier processing near the data (**R1**).

Since the underlying technology stack was different and there was no need of transactional persistence model a new approach was also chosen for the web application. It makes use of a popular web application framework, called PlayFramework[9], which does not rely on a complex technology, such as EJBs. It supports the developer to follow the MVC pattern and allows the rapid development of a REST [13] interface. REST was chosen for two main reasons. For one, it is easy to understand and pretty straightforward to implement. This allows easy integration with other tools such as monitoring services and a variety of client applications. The second reason is that other technologies such as JavaServer Faces Technology[10] (JSF) and Enterprise Java Beans Technology[11] (EJB) would have meant that only a few application servers would be compliant and able to support the application.

The UI was implemented with the help of the framework, by utilising Scala[12] templates and standard technology such as HTML5, Javascript and CSS3. It enables the user to select her collection and obtain a deeper overview by filtering the data based on specific properties and values as shown in Figure 4.3. The screenshot shows the overview page of a real world collection. It shows the distributions of seven properties: mime type, format, format version, validity, well-formedness, creating application and created date. In the upper right corner of the overview, the application provides statistical information about the size of the objects (**R4**). Although it is not possible to see from the screenshot, the diagrams are interactive and give the user the possibility to filter the data based on any measure (**R5**). From the navigation bar in the top, one can see that C3PO also has features for browsing the objects, finding representative sample records and exporting data (**R3**).

[9] http://playframework.org

[10] http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html

[11] http://www.oracle.com/technetwork/java/javaee/ejb/index.html

[12] http://www.scala-lang.org

Figure 4.3: Screenshot of C3PO - Overview of a collection.

Comparison and Results

Due to the changes made in the second iteration significant performance and scalability optimisations were achieved.

Table 4.1 shows the average time needed by C3PO for processing 1000 FITS files in each of the two iterations. This includes traversing, reading in the data, parsing it and storing it within the data store. All of these experiments were conducted on a single commodity machine.

The collections used are from real world preservation scenarios. Both are significant in size, where the first consists of about 42 thousand PDF documents and the second consist of about 550 thousand web harvested material. The processing includes the time for parsing, converting to the internal data model and storing to the data base. Both experiments are conducted on the same common hardware.

The first implementation needed 350 seconds per thousand FITS files on average for the PDF collection and didn't finish due to memory issues for the web collection. In the second iteration the memory issues were resolved and the DOM parser was exchanged by a SAX parser, which resulted in 80 seconds per thousand files on average for the document collection and 98 seconds per 1000 files for the web data. After the removal of the ORM and some further optimisations in the SAX parsing approach a significant boost was achieved; the PDF collection was processed with a rate of 3 seconds per thousand files and the Web data with 2 seconds per thousand files.

Collection	**1. Iteration**	**2. Iteration**	
	DOM + ORM	**SAX + ORM**	**SAX + Mongo**
PDF collection	350 sec	80 sec	3 sec
Web collection	threw exception	98 sec	2 sec

Table 4.1: C3PO's average processing time of 1000 files in two different collections

The process parallelisation improved the speed even more. Table 4.2 shows more than 30% speedup for processing the whole govdocs1 collection, which consists of about 1 million FITS files of mixed data.

Collection	**1 Thread**	**8 Threads**
govdocs1	165 min	104 min

Table 4.2: C3PO's processing time of the govdocs1 collection with one and 8 worker threads

Furthermore, exporting a sparse matrix of all property values for every element in the collection was shown to be much faster. This was due to the fact that the internal representation of the data needed no JOINS anymore and can be done by single iteration over a database cursor.

Many of the analysis queries were changed to be done by map-reduce jobs. Since these run near the data and not in the application itself, the scalability of such queries was significantly improved.

Chapter 5 summarises detailed performance measures of the govdocs1 and a web archive collection as well as the time needed for profile generation and matrix export and gives an overview of the capabilities of the tool.

Interfaces and Extension Points

In order to extend the system, the framework provides a couple of interfaces. One important extension point for later use is the *GathererInterface* which provides an abstraction for the source of the raw meta data. It exposes a unified interface allowing the *Controller* to obtain streams to the next N files that have to be processed. This design allows a transparent view to the other modules in the system. The prototype of C3PO provides an implementation only for local file systems. However, extending it to fetch data from a different source is just a matter of implementing a single interface, which is able to count the files to be processed and to open streams to the next N files. It is up to the implementing class to decide, whether the data will be retrieved over the network and the stream will be passed directly for further processing or it will be cached in batches to the local file system. Depending on the use case both could make sense and thus it is left in the responsibility of the service provider.

Clearly, meta data representation is another important aspect in such as system. For the prototype, FITS was chosen due to it benefits regarding normalisation and conflict detection. Nonetheless, other formats can make sense in specific use cases and when integrating with different sources, that utilise different meta data schemas. In order to extend C3PO, one has to implement a simple adaptor that is able to parse the new schema and return the data in a way that fits C3PO's data model. A drawback here is that this will potentially break the property normalisation. For example, two different meta data schemas can have two different property names with the same semantical meaning. There are a couple of solutions to this problem, which are taken into account in the design, but are not implemented due to insufficient information of real world scenarios. The first could be to provide the mapping between these colliding properties via some user configuration and to take them into account during parsing or post-processing. The second solution would be to allow the usage of a single adaptor based on the use case.

In order to obtain a profile, client applications can use the REST API. With a few simple calls, a XML representation of a collection profile can be generated and retrieved. This file conforms to the proposed schema in Chapter 3 and provides simple aggregations of the data.

4.3 Representative Sets

Representative sets are one of the most important features of C3PO, as they provide the basis for experiments during the planning phase. The selection of valid representatives can influence the decisions of a planner for the future of specific content strongly. There are many different ways of selecting a small set of representatives. Here we present a few algorithms that were implemented and discuss their benefits and drawbacks. It is important to keep in mind that real world scenarios can involve large collections of thousands to millions and even more digital objects, but usually the experiments during preservation planning are done on a very small set of representative sample objects in the order of 10 objects.

In the following, we present only the core of the algorithm and leave out some basic input checks, such as the initial size of the collection or the filtered subset, etc.

Random Selection

As the name suggests, this algorithm takes N random elements from the larger set and returns them. A simple pseudo code implementation looks like Algorithm 4.1. Unfortunately, this approach is widely used in real world scenarios, due to the lack of automation support and understanding of the content at hand. One drawback of this approach is that it will most probably provide terrible results when applying it on highly heterogeneous content. Nonetheless it could be useful in some cases, if the content is first filtered. For example, one can first apply some filters with C3PO and split the collection into smaller sets that are homogeneous with respect to certain properties and then apply this algorithm on each of the subsets. If the smaller subsets are homogeneous enough, it is possible to achieve good results. Nonetheless, splitting and analysing the content could potentially take a lot of time.

Input : A set of digital objects S and a upper limit N for the sample objects set
Output: A set of random representative sample objects R

```
   // gets the number of objects in S
1  count = S.size();
2  while R.size() <= N do
       // gets a pseudo random number between 0 and count
3      rand = getRandomNumber(count);
       // S[rand] get the object at index 'rand'
4      sample = S[rand];
       // add() appends the object to R
5      R.add(sample);
6  end
```

Algorithm 4.1: Random sample selection

Size Statistics

This approach is probably the most common approach currently used by planners and preservation experts. It also selects random elements, however it considers some statistical information regarding the size of objects. This decision stems from the observation that preservation action tools often perform badly on objects with a large variation in size. Thus planners often take the smallest, largest and several average-sized objects and conduct the experiments over them. If there are no other significant variations and differences in the objects, the selected representative set, could provide good results for planning. However, if the objects have significant variations in characteristics, other than the size that might influence the preservation action tools and the algorithm would be error prone.

A possible pseudo code implementation is provided in Algorithm 4.2. As this algorithm makes use of the standard deviation in order to find objects of size near the average size, it utilises an implementation of another online parallel algorithm for calculating the variance of a set of data. An online algorithm is one that operates on the data without having all needed input from the beginning. This particular calculation of the variance in a single pass over the data was

suggested by West in [47] and is a parallel algorithm that takes into account a sum of weights at each step. Listings 4.2 to 4.4 show the map, reduce and finalize functions (in JavaScript notation) of a Map Reduce job that implements West's algorithm.

This approach is slightly better than the previous one, as it considers at least one characteristic of the meta data, that indeed often has influence on the experiments. If applied on a homogeneous set (with respect to the digital object type), it could give good results. Nonetheless, there are other factors that have to be considered, especially in cases where the variance of the size in the collection is small.

Input : A set of digital objects S and an upper limit N for the sample objects set
Output: A set of random representative sample objects R

```
   // gets the number of objects in S
 1 count = S.size();
   // map reduce job that calculates
   // statistics for the property 'size'
 2 stats = numericMapReduceJob('size');
 3 min = stats.min;
 4 max = stats.max;
 5 avg = stats.avg;
 6 sd = stats.sd;
 7 low = floor(avg - sd / 10);
 8 high = ceil(avg - sd /10);
 9 minObj = querySize(min);
10 maxObj = querySize(max);
11 A = querySizeBetween(low, high);
12 R.add(minObject);
13 R.add(maxObject);
14 while R.size() < N do
15 |   R.add(A.remove(0));
16 end
```

Algorithm 4.2: Size Statistics Sample Selection

```
1  function map() {
2    var size = this.metadata['size'].value;
3    emit(1,
4      {sum: size,
5     min: size,
6     max: size,
7     count:1,
8     diff: 0,     // sum of squares of differences from the (current) mean; sum((val-mean)^2)
9     });
10 }
```

Listing 4.2: The Map function for basic statistical aggregations.

```
function reduce(key, values) {
  var a = values[0];    // will reduce into a
  for (var i = 1; i < values.length; i++){
    var b = values[i];    // will merge b into a
    var delta = a.sum / a.count - b.sum / b.count;    // a.mean - b.mean
    var weight = (a.count * b.count) / (a.count + b.count);

    a.diff += b.diff + delta * delta * weight;
    a.sum += b.sum;
    a.count += b.count;
    a.min = Math.min(a.min, b.min);
    a.max = Math.max(a.max, b.max);
  }
  return a;
}
```

Listing 4.3: Th Reduce function for basic statistical aggregations.

```
function finalize(key, value){
  value.avg = value.sum / value.count;
  value.variance = value.diff / value.count;
  value.stddev = Math.sqrt(value.variance);
  return value;
}
```

Listing 4.4: The Finalize function for basic statistical aggregations.

Systematic Sampling

Systematic sampling is an optimisation of the random approach and is an equal-probability method. In other words, it is fairer as every object in the set has equal probability to be chosen as a representative. It divides the content into *n* buckets and calculates a *skip* variable by dividing the number of objects in the population **N** by the number of buckets **n**. Then a random starting point is chosen between *zero* and the calculated *skip*. Afterwards, adding the *skip* to the index of the last chosen element chooses the next *n-1* elements. This will result in one element per bucket.

Even though this method is fairer, it still shows the same problems as the random sample selection algorithm. A pseudo code implementation is given in Algorithm 4.3

Input : A set of digital objects S and an upper limit N for the sample objects set
Output: A set of random representative sample objects R

```
   // gets the number of objects in S
1  count = S.size();
2  limit = round(count / N);
   // generates a random number with a upper limit
3  skip = nextRandom(limit);
4  while R.size() < N do
5  |   offset = skip * R.size() + skip;
6  |   R.add(S[offset]);
7  end
```

Algorithm 4.3: Systematic Sampling Selection

Distribution Coverage

The distribution coverage algorithm tries to find sample objects with the same distribution of the property value pairs of characteristics chosen by the planner. For example, if there is a collection consisting of *40% PDF documents* and *40% Word documents* and *20% other documents*, the desired sample set size should be ten and the chosen property is *format*, then the algorithm will select four *PDF files*, four *Word* files and two other files at random that are not PDF or Word documents. If the planner also wants another property value distributions taken into account, then these are considered by building all possible key-value pairs. The algorithm tries to find the nearest distribution possible over all the combinations of the selected properties and their values. Continuing the previous example, If a planner wants to consider also the property *format version* and the collection has documents with versions *1.2* and *1.4* for PDF as well as *2003* and *2007* for the Word documents, then the algorithm will find all possible combinations *(PDF - 1.2, PDF - 1.4, PDF - 2003, PDF - 2007, Word - 1.2, Word - 1.4, Word - 2003, Word - 2007 etc.)*. Afterwards the occurrences for each combination in the data set will be counted and the approximate distribution will be returned. This means that the number of combinations is equal to the product of the distinct value occurrences for each property. In this example, two distinct occurrences for the property *format* and four distinct occurrences for the property *format version* equals eight combinations. Obviously some of the combinations will always return zero, as there is no such format as *Word - 1.4*.

This can be avoided if enough information about the correlation of the different properties is provided. As this is not the focus of this thesis, the current version of the algorithm builds all possible combinations and does not consider the correlation between properties.

Algorithm 4.4 presents a simple pseudo implementation. It takes a set of digital objects S, an upper limit for the samples set and a set of properties P.

In a first phase a simple multi-dimensional array (matrix) is build. It goes over all passed properties and their distinct value occurrences. In order to populate it the algorithm obviously needs to issue database queries (line 5) or receive the distinct property value pairs as input.

In a second phase, it creates all possible combinations of the property value pairs as in the example above. Each combination consists of one value per property and the count of objects

that have these exact values for the given properties. In the next step the combinations are sorted based on count.

Afterwards the algorithm iterates over the combinations until the size of the set of sample objects R is smaller than 'N' and calculates the ratio of count of objects in each combination and the total collection size. It issues a query to collect a number of objects that have the same ratio in the sample set R as the ratio within the collection.

This algorithm is better with respect to heterogeneity of properties within the same digital object type. A planner that makes use of it has to understand that the algorithm does not consider the long tail of files. It is suggested, that these are handled separately, as they are often responsible for messed up experiment results.

Nonetheless, the selection of a larger set of properties could impede the performance and the validity of the results, as the occurrences for each combination of property and values will get significantly smaller. Thus a planner has to consider carefully, which combinations of properties make more sense.

4.4 Future Points of Interest

There are many topics that can be handled and implemented in future work. Here we outline some and briefly discuss them:

- *Scalability* is very important, as the volumes of data will grow more and more over time. Even though the hardware resources and provided performance also grow over time the issue of scalability is very important. The author believes that horizontal scalability is the best way to pursue. The current design decisions follow that path. More specifically, future enhancements should include distributed map-reduce jobs and the caching of specific results, which will enhance the systems responsiveness.

- *Continous Profiling* is important as collections change over time. The support for including meta data of new and changed objects as well as adapting the computed aggregations in a easy, fast and scalable fashion is critical for the success of such an application in production use. Thus the investigation of the problems that arise with this will be important for future versions of the tool. The Map-Reduce facilities provide a potential solution to such a problem. In contrast to SQL solutions, here it is possible to aggregate new data and reuse old results, which are then just merged as the reduce function is always the same.

- *Input to Digital Preservation Tools*. Monitoring systems and simulators enhance the preservation planning processes by providing relevant and trustworthy information that is often hard to obtain due to its distribution and by offering simulations and projections of different outcomes based on current decisions and policies. Such systems heavily rely on larger amount of data. Integration with a content profile tool will play an important role for such tools as well as preservation planning activities.

- *Representative Sets* are the foundation for unbiased experiments and thus better performing algorithms in terms of speed and effective selection should be the focus of future work as well.

Input : A set of digital objects S, an upper limit N for the sample objects set and a set of properties P

Output: A set of random representative sample objects R

```
   // creates a matrix M with the properties and all distinct
      values for each property
1  M[P.size()][];
2  i=0;
3  foreach property p of P do
4  |  j=0;
   |  // issues a database query
5  |  DV = findDistinctValuesForProperty( P );
6  |  V[DV.size()];
7  |  foreach value dv of DV do
   |  |  // PV is property value pair wrapped in an object
8  |  |  V[j] = PV(p, dv);
9  |  |  j = j+1;
10 |  end
11 |  M[i] = V;
12 |  i = i + 1;
13 end
   // C is a list of combinations over all property value
      pairs
   // A combination is a wrapper object that has a distinct
      value for each property and the count of objects that
      conform to the combination
   // The size of the combinations equals the product of the
      size of distinct values for each property
14 C = combinations( M );
   // sort based on count
15 sortDescending( C );
16 foreach combination c of C do
17 |  if R.size() < N then
18 |  |  percent = c.count() * 100 / S.size();
19 |  |  tmpLimit = round(percent / 100 * N);
   |  |  // get maximum 'tmpLimit' objects out of S that
   |  |     conform to the current combination
20 |  |  R.addAll(getObjectsConformingToQuery(c.query(), tmpLimit));
21 |  end
22 end
23 return R;
```

Algorithm 4.4: Distribution Coverage Sample Selection

- *Visualisation and Interactivity*. A wider variety of visualisations of different aspects of a collection can help and influence the decision of a user.

4.5 Integration

The prototype was integrated with two digital preservation systems developed within the SCAPE project. Here we shortly describe the concept of the integration, why it makes sense and how it contributes towards a full implementation of the presented preservation planning environment in Chapter 3.

Scout - A preservation monitoring system

As preservation planning is a continuous process that has to be repeated constantly, the preservation environment foresees a special component that monitors certain types of changes in characteristics that are of interest to a planner. Monitoring is responsible for the detection of important events such as new emerging formats, new software or new versions of software, but also violation of institutional policies and many more. One of the important aspects that a monitoring component has to follow is the content profile. Doing so, it can cross match the formats within the profile with formats present in registries and institutional objectives. For example, an institution might have an objective that all image content within a repository has to have lossless compression.

As C3PO exposes a REST interface for generating and retrieving a profile in XML format, integration between the SCAPE monitoring Service - Scout, and C3PO was created. Scout contacts the profiler over the REST API and periodically fetches a profile for each known collection. Afterwards the profile is parsed and the values of specific properties such as size, format distribution, etc. are collected.

If a planner has created specific conditions or constraints within Scout and these are met or violated after the new values are measured, then the planner will be notified and will potentially have to change the current plan for a given collection.

Plato - A preservation planning suite

A huge part of the planning environment is covered by the preservation planning tool - Plato. Plato supports a planning expert through all phases of the planning process, from plan creation to building an executable plan .

Up until the third version of Plato, a planner had to define the scope of every new plan by hand. This implied that the definition of the preserved collection was done manually. As this is a tedious process, planners often provide only high-level descriptions of the collection and also select all sample objects randomly.

Throughout the SCAPE project, C3PO was integrated with Plato in order to semi-automatically define the scope of a plan not only in a high level assertion, but also in more detail, providing a correct format distribution and more. Allowing a planner to filter a collection and export an XML profile via C3PO enables the expert to easily start creating new plans, without having to manually inspect the type of the objects and find representative.

After the exported profile is uploaded to Plato, the tool parses it and fills in all necessary information. Moreover, the sample records information is filled automatically and the used algorithm is provided in order to have evidence later on. As every object known to C3PO has a unique identifier (within its source), Plato can check where the collection comes from. If the source is known and the user provides the correct credentials, Plato automatically downloads the sample objects for the experiments during the following steps. Having a list of object identifiers that are part of the collection has also another advantage. After a plan is build and ready for execution, Plato can reopen the stored profile and read out all identifiers that are part of the collection. These are included within the executable plan. This has the advantage that once the executable plan is fed back to the source (repository) for execution, the repository automatically knows on which objects it has to apply the executable actions defined in the plan.

Observation

This rather simple integration of profiling with the two larger components of the preservation environment enhances the experience of a planner greatly. A preservation expert does not only benefit from the automation support provided by the profile, but also from the fact that now he can do a complete preservation lifecycle of a collection. To illustrate this example, consider the following scenario. In a first step the institutional policies define that objects within the repository have be covered with a plan. An expert creates a profile with C3PO and uploads it to Plato. Plato automatically defines the scope of the plan, fills in the sample objects and automatically fetches them from the repository. The expert defines and conducts the experiments with help of Plato. After that a recommendation is chosen and an executable plan is created. This plan is deployed within the repository and gets executed. Meanwhile a monitoring component observes the profile of the repository as well as the operations of the execution and notifies the planning expert when certain changes or potential problems occur.

CHAPTER 5

Evaluation

This chapter summarises the outcomes of two case studies conducted with the implemented prototype C3PO and their results. First the goals of the experiments and some general information about the content sets are defined. Then the two case studies are summarised and some of the interesting results are presented. The first use case is conducted with format heterogeneous content and the second one over a web archive.

5.1 Goals and General Information

In order to test and validate C3PO, a large set of files is needed to conduct experiments over it. For this work we use two sets of data to conduct similar experiments on the same commodity machine. The number of objects in each set is in the order of hundreds of thousands objects with a upper border of one million. This volume is large enough to capture the requirements of many organisations and institutions. However, it is noteworthy, that there are institutions (such as web archives) with many millions and even billions of objects. Future work might focus on such case studies.

The machine used for all the experiments (unless otherwise stated) is a common laptop with a 2.3 GHz Intel Core i5 Processor (2 Cores), 8 GB RAM and a common internal hard drive with 5400 rpm. As C3PO is meant to run on a server, it is very likely that this configuration is much less capable than common server used within stake holding institutions, considering the current trend of hardware technology.

The author's hypothesis is that the processing power as well as the hard drive device would be the limiting factors during data gathering in the following experiments. The processing of each file alone is a fast operation, however traversing the file system, opening and closing a stream to each file and storing the data to the local hard drive (within the database) are relative expensive operations. Thus the disk write speed and the processing capacity are of a bigger importance then the RAM during the initial phases of the evaluation. The RAM capacity will play an important role during analysis, as the map-reduce jobs will strongly depend on that.

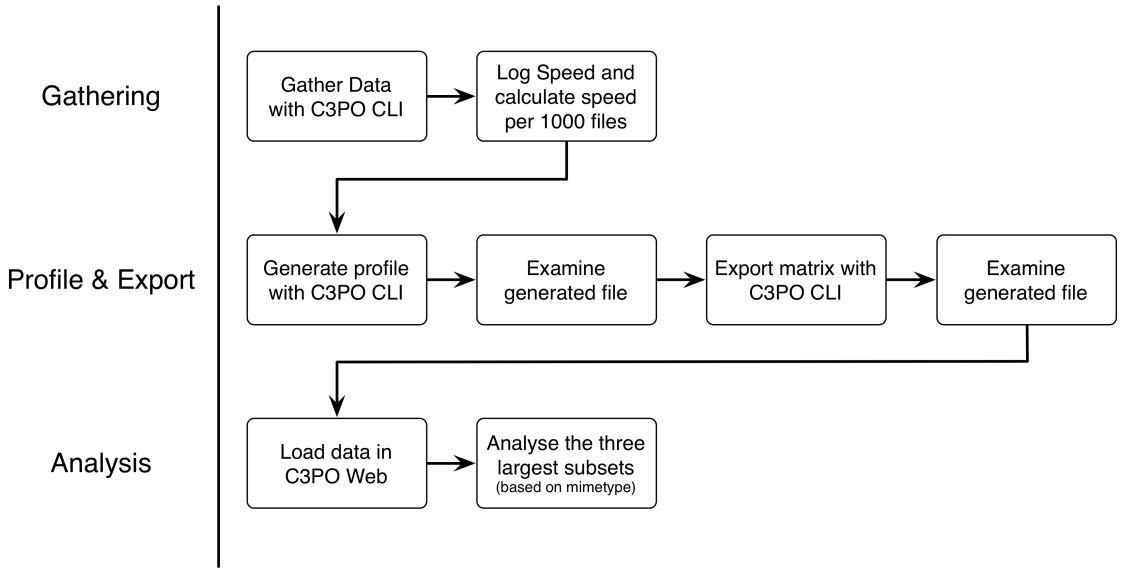

Figure 5.1: The steps for each case study

The goal of these case studies is to find out the usefulness of the tool in terms of speed, scalability of resources and volume of the data as well as to find out its limitations and places for enhancement and optimisation in future versions.

It is important to note that the usefulness in terms of preservation planning will not be validated and is not in the scope of the following experiments and observations. One way to do this would be to do the whole preservation planning procedure for each of the partitions created by C3PO. In a next step representative, samples have to be chosen and then the recommended action has to be applied over the whole partition. In a last step, special quality assurance processes will have to validate the results of the planning process. If they were successful, this will be a hint for the usefulness of tools such as C3PO. Since this will require many resources and there are huge gaps in terms of quality assurance possibilities on larger scale, it will be rather hard to conduct such a case study successfully.

The actual experiments conducted in each case study follow the steps of the flowchart in Figure 5.1. All of them used C3PO v0.2.0 that can be downloaded from here (`https://github.com/peshkira/c3po`).

In a first step, the FITS data is gathered with the C3PO CLI application into a local instance of a MongoDB document store and the time is tracked. Afterwards the time of the whole operation is logged as well as the time for processing 1000 files is calculated. We conduct this part of the experiments three times with different configuration of the application in order to test the parallelisation speedup.

In a second step, we generate a XML profile with the command line application, which gets examined for errors. The size and the usefulness of the generated file are also evaluated considering the integration with other tools such as Plato and Scout. In the same step, we also

generate a .CSV file containing a matrix view of all gathered data. Consequently, we examine the produced file.

In the last step we load the data within the C3PO Web application and after giving an overview of the data we analyse the three largest subsets of the collection based on the mime type.

Conducting all these steps ought to give the reader an overview of the strengths and weaknesses of the prototype implementation and what would be possible with a real world application.

5.2 GovDocs1 Documents

DigitalCorpora.org[1] is a website that provides digital corpora for use in computer forensics education and research. The site offers different file sets, network dumps, disk images and more. For the following experiments we use the GovDocs1 file set, which contains of nearly one million freely redistributable files that resided in the .gov domain. The files are randomly distributed into 1000 directories with up to 1000 files in each directory and can be downloaded at `http://digitalcorpora.org/corp/nps/files/govdocs1/`.

Data Description

Forensic Innovations Inc.[2] have provided a simple statistical report that shows some of the important characteristics of the set, which we will try to find out with C3PO and provide even more deeper insight. The report can be found here: `http://digitalcorpora.org/corpora/files/govdocs1-simple-statistical-report`. In Table 5.1 general information such as file size and volume is summarised, whereas Table 5.2 provides a summary over the content of the different files. Note that the total sum of files in the second table is more than the number of files in the set, meaning that many files are counted twice or even more, which is not very helpful for digital preservation activities.

Characteristic	Total
Nr. Files	986278
File Size (KB)	488658258
Wrong File Extension	33917
Scan Time	10:12:37

Table 5.1: General information of the GovDocs1 data set.

[1] http://digitalcorpora.org

[2] http://www.forensicinnovations.com/

Content	Total
Personal/User Data	961914
Text	727217
Document	539100
Hypertext	467405
Graphic Image	464870
Macro/Script	351781
Font	231275
Spreadsheet	85110
Program Data	41616
Source Code	36580
Raw Printer Data	26190
Database	24820
Archived Files	14093
Video	3483
Email	2007
N/A	882
Template	306
Program Executable	277
Presentation	222
CAD/3D Model	138
Game Data	15
Sound/Audio	10
Shortcut/Link	5
Library of Functions	2
Form	2
Encryption Key	1

Table 5.2: Content types within the GovDocs1 set as the preliminary data shows.

Experiment Preparation

In order to conduct the experiment, all the files have to be characterised with the FITS tool. In order to automate this task, the authors have used a workflow engine developed by the University of Manchester called Taverna[3]. With the help of Taverna, one can create parallelised workflows that consist of number of small steps and tasks. In this instance, the files were copied via *scp* from a storage server, FITS was executed in parallel on each of the files and the output was stored on the experiments machine. A screenshot of the produced workflow is presented in Figure 5.2.

Unfortunately, the current version of FITS (v0.6) was not stable for all file types and thus it was unable to produce output for some files. Thus all the experiments conducted on the set are

[3] http://www.taverna.org.uk/

done on a slightly smaller subset consisting of 945746 instead of 986278 files. The total file size of the FITS meta data was 5.37GB.

Disclaimer: Due to the unpredictable behaviour of FITS on certain file types, the workflow had to be restarted numerous times. Thus it was impossible to capture the real time for characterisation. In the following, all time intervals that are given are only for the execution of C3PO, unless otherwise noted. Thus any comparison with the scan time in the preliminary data should not be taken lightly. Nonetheless, the results provided here shall give a good overview of what would be possible with a tool such as C3PO.

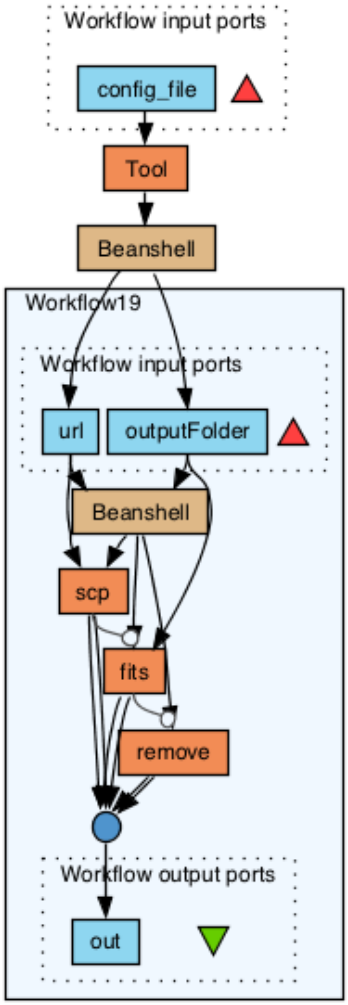

Figure 5.2: The taverna workflow used to run FITS on the govdocs1 collection.

Gathering

In the initial test the gathering was conducted in a single thread environment. The following command line shows the executed C3PO command. The *'g'* option defines to the location of the FITS files to gather, the *'r'* option starts a recursive file system traversal and the *'c'* option sets a name of the collection.

```
c3po -g ~/path/to/folder/ -r -c govdocs1
```

The processing of all 945746 files took more than 167 minutes. 47 files were not processed successfully due to malformed meta data files, which were caused by failures during the execution of the FITS tool. After initial preparation and setup, processing one thousand files took 10.46 seconds on average.

The same procedure was conducted with four and 8 threads. The speedup was 121 and 108 minutes respectively. The average processing of one thousand files was 7.54 and 6.73 seconds respectively. This shows more than 35% increase in speed. The results are summarised in Table 5.3

Files count	1. Thread	4. Threads	8. Threads
945746	167 min	121 min	108 min

Table 5.3: C3PO's gathering performance of the govdocs1 collection.

More powerful hardware (with a solid state drive disk and faster CPU) will most probably enable a speedup up to 50% in comparison to the single threaded solution. A next step of improvement towards the next threshold could be the parallelisation of the process on different nodes against the same document store and even distribution the store itself. These improvements are not included as part of this work, but are possible next steps, which will enhance the process even further.

Profile & Export

In this part of the experiment C3PO was used to generate a profile of the data processed in the previous step and to also export it. The generated files are then inspected. The following command line shows the executed C3PO command. The *'p'* option defines to the location where the profile should be created and the *'c'* option specifies the name of the desired collection.

```
c3po -p ~/path/to/output/folder -c govdocs1
```

The time taken for the profile generation of the whole govdocs1 collection was 12 minutes. The profile included 112 properties. The output file was 53 KB in size. Considering that content profiling is not meant to be a process that delivers results instantly, these preliminary output seems to be feasible and is acceptable considering the integration with other tools over a network.

In the case that the aforementioned improvements in terms of processing are successful, then it is likely that the generation of a profile over larger content will take more time. If the

time of profile generation takes too long in such a case, a new polling strategy will have to be implemented, where an asynchronous profile generation job is submitted and the status of the result is polled by the client.

The same command was then executed with the *'ie'* switch, which includes the element identifiers in the profile. The time taken was 12 minutes once again and the resulting file was 60 MB in size. This was due to the fact that all object identifiers were included in the profile. Since the collection is of considerable size, this shows how infeasible this option is, especially for large-scale usage. It is helpful for demonstration purposes on smaller sized collections, but will most probably not be feasible in real world scenarios. This part of the profile could be replaced by a special query that selects the digital objects falling into this profile. This presents a potential problem, as the query should be agnostic to the data representation, but expressive enough to select all of the matching objects. Furthermore, any client should understand this query and integrating applications, such as planning tools and digital repositories, as it will be the interface between these. If this is can be achieved, the footprint will be once again rather small. A potential solution for this problem could be achieved by making use of the Search and Retrieve by URL - SRU specification as presented in Section 3.2 on page 3.2

Afterwards all the data was exported to a .csv file in a sparse matrix, with the object identifiers as rows and the properties as columns. The following command line shows the executed C3PO command. The *'e'* option defines to the location for the exported file and the *'c'* option sets a name of the collection.

```
c3po -e ~/path/to/output/folder -c govdocs1
```

The generation took a little more than 2 minutes and resulted in a file of 430 MB in size. This matrix view is very helpful to obtain an overview of the whole collection and can be used to apply some more complex filters that are not possible with the web application. On the downside, it is questionable if this approach will scale on larger content. While state of the art spread sheet processors still cope with files of this size, it is highly questionable if it will be possible to open and process a much larger file. A solution for this potential problem would be to split the exported matrix into smaller pieces and process them separately.

Analysis

In the last step the web app of C3PO was used to look at the govdocs1 content set and obtain an overview. The following describes the set and what was possible with the tool.

C3PO revealed that the whole collection of digital objects had an overall storage size of 447.36GB of data. This value seems to be realistic considering that FITS failed on numerous objects and the C3PO gathering process failed on some as well. The smallest object has a size of 7 Bytes and the largest - 1.52GB. On average all of the objects have a size of 0.48 MB with a standard deviation of 4.29 MB. On the downside, the application did not allow the selection of the smallest and largest objects, which could be easily fixed in a next version. In the current version, this will be only possible via the .csv export.

C3PO immediately showed that the collection consists of 46 different mime types and in addition there is one subset of conflicted mimes, which is the second most occurring. The first

9 most occurring mime types represent nearly 80% of the whole collection and are presented in Figure 5.3. The correlation between the mime type and the format of a digital object implies that both distributions are similar. The generated distributions of C3PO validated the correlation as well.

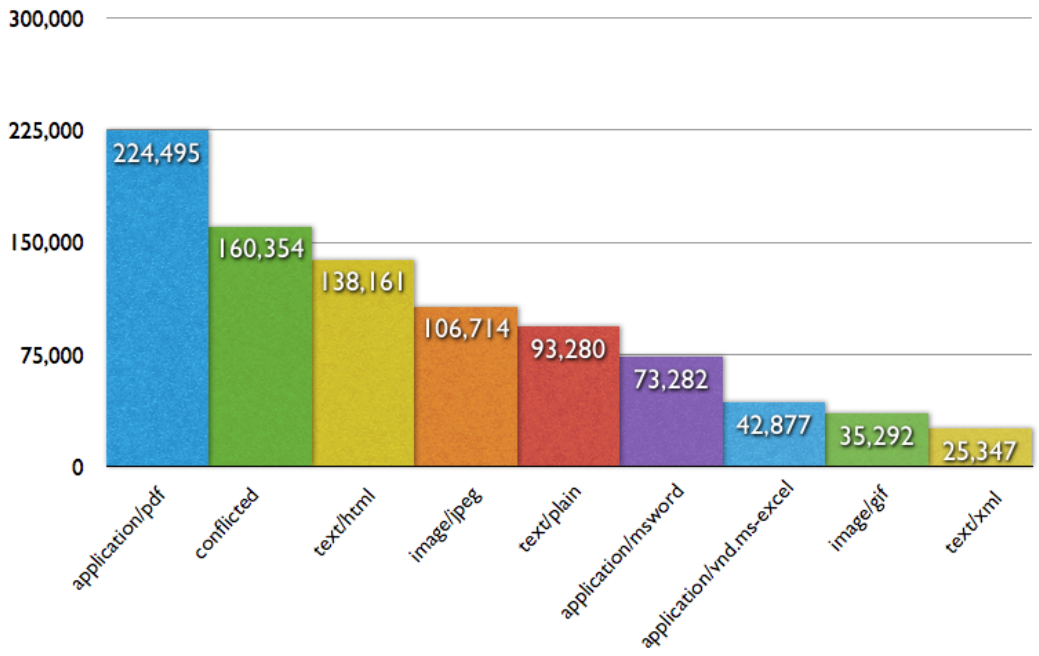

Figure 5.3: The 9 most occurring mime types within the GovDocs1 set as C3PO showed.

The next interesting observation was the creation date distribution of the content set. The following Figure 5.4 presents it. The total of the files in this distribution is much less than the total of the collection, because of the data sparsity. Nonetheless, there are some interesting observations that can be made out of this distribution. Firstly, the objects created between 2003 and 2008 are much more than the rest. This can be related to the fact that data production does not increase constantly, but rather exponentially during the years, but this conclusion might be biased as the data may not be sufficient to back it up. Secondly, it is very interesting that the data created in 2009 is less than some data created during the 90s. Thirdly, there is one small subset that is gathered in year '-1'. This is clearly a faulty measure provided by some of the characterisation tools bundled in FITS, which proves the point of the importance of meta data quality. Last but not least, one subset was created in 1910, which is rather peculiar and is most probably related to bad data quality.

Portable Document Format Adding a new mime type based filter showed that the 224495 application/pdf files in the govdocs1 set consisted of three different PDF formats (PDF, PDF/A

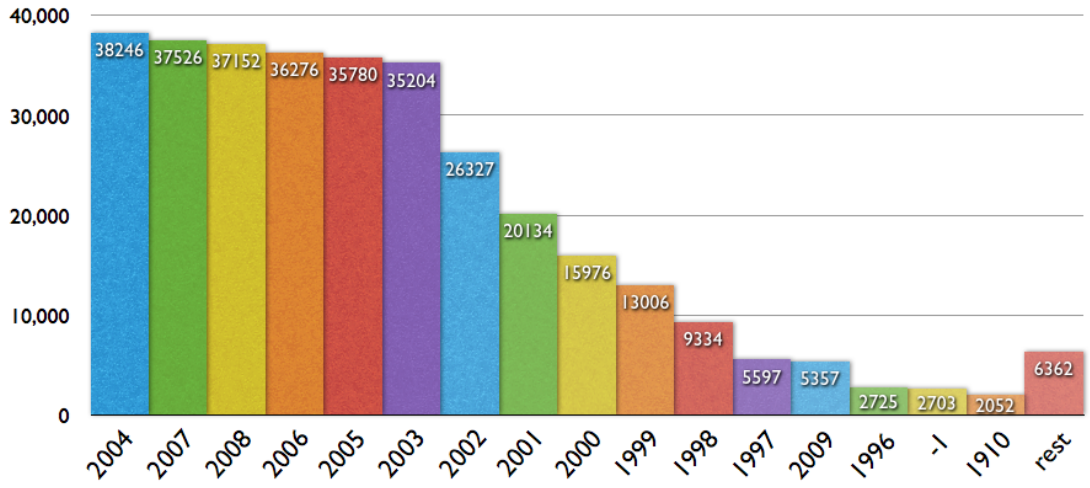

Figure 5.4: The creation date distribution of the govdocs1 collection.

and PDF/X) with more than 10 different versions. This is about 28% of the storage size needed for the whole collection.

In a next step, the validity and well-formedness of the documents was examined. 90% of these documents were valid, nearly 10% were invalid and less than 1% were unknown. Almost all documents were well formed. About 1% was not and once again less than 1% was unknown. Considering only files with unknown validity, made it possible to conclude that the unknown files in terms of validity and well-formedness were exactly overlapping. Because of this fact, they are excluded of the following observations.

C3PO showed that all valid PDF documents were also well formed. However, 8% of all well-formed PDF objects were invalid. None of these 8% were in format PDF/X, which implies that all PDF/X formatted objects were reported to be both valid and well formed.

Selecting the subset of invalid and not well-formed documents, which were about 1% of the whole PDF collection, showed that it consists only of PDF documents (in all versions from 1.0 to 1.6). C3PO also provided a list of so called 'evil' applications that created these malformed files. Most were created by different versions of Acrobat Distiller.

Finally the distribution of the properties 'is rights managed' and 'is protected' were generated. It turns out that only 81 PDF documents had rights data associated with them, whereas 97% didn't have any. For about 2% it was unknown. The protected PDFs were many more in comparison - about 4%. Again 2% were unknown and the rest were not protected. Eight objects were conflicted in terms of protection. All of these were invalid and not well formed. Two had the format PDF 1.4 and six - PDF 1.6.

The analysis of this type of objects was quite easy and the tool enabled the user to drill down and find out interesting facts about this subset.

Conflicted Examining the objects with conflicted mime type values with C3PO proved to be rather hard and showed room for optimisation and enhancement in the future versions of C3PO. Besides of the storage size statistics the information was not very helpful. The storage size of all 160353 objects was about 81GB. All the formats in the subset were also conflicted and only versions were shown, which was not enough information to gain an overview of the conflicted subset (the second highest in the whole set). An idea was to take a look at the creating applications and to obtain a rough overview of the type of documents in this subset. However, 99% of these were unknown.

In terms of validity and well-formedness, the following observations were made. For 48% of the subset both these properties were unknown. All 20% of the valid objects in this subset were also well formed. The other 32% of invalid objects contained both well formed and not well formed objects in ratio 3 to 2.

All this showed, that it would be helpful to a planner to see a list of conflicted values or some kind of weighted distribution. Additional filters over these would also be helpful. Furthermore, it will be beneficial to apply special rules that resolve the conflicts in cases where the characterisation tools provide conflicts for similar formats (e.g. text/xhtml, text/xml).

JPEG As there is a second case study, conducted over data from a web archive that will naturally include many html files, we skip the third most common mime type (text/html) in this collection and focus on the next one - image/jpeg.

This subset consists of 106714 objects or about 11% of the whole set. The storage space needed for this subset is about 34GB.

Two formats were identified within this subset (JPEG and EXIF) with more than 10 different versions. 80% of the files were valid and well formed. Only 3 objects were invalid. The rest was unknown both in terms of validity and well-formedness.

The JPEG files presented 95% of the subset and most of them had YCbCr colorspace. The rest were unknown.

The EXIF files were significantly smaller (a bit more than 4%). For most of them the colour space was unknown. The rest were RGB-coloured images.

Other interesting data provided by FITS were the GPS coordinates of some images. Unfortunately, C3PO is not able to make use of these in order to visualise them. Nonetheless, an overview of the objects having such meta data could be an interesting asset in a digital preservation activity considering a scenario where such meta data has to be kept after a preservation action is conducted.

5.3 Web Archive Data

The Danish State University Library[4] (SB) has a mandate to maintain the whole Danish web archive. This includes every website of the Danish domain (.dk top-level domain), all web content hosted by Danish companies, all content produced by Danish and more. Currently the

[4] http://en.statsbiblioteket.dk

archive has content with more than seven billion objects with size in the order of petabytes. The growth of the archive is expected to rise exponentially during the next decade.

Data Description

In collaboration with the University of Technology in Vienna, the SB has shared the FITS metadata of a subset of the web archive harvested over the last eight years. The files were selected randomly and contain not only HTML documents but also all related content, such as stylesheets, script files, image material, etc. Due to the volume of the data, even the administrators of the archive do not have an overview of the data they possess. In the following we analyse 958,953 FITS files with an overall size of 4.18GB.

Experiment Preparation

In this case, the SB generated the FITS files. They used an extended version of FITS, which was provided by the University of Technology in Vienna. This version deactivated the JHOVE tool for html files, due to its known bad performance and included another tool that provides some more information about html content. Once again, all time measurements provided in the following solely include the time that C3PO needed for execution, unless otherwise stated.

Experiment

Here we undergo the same steps as with the previous experiment and measure the times needed for gathering, processing and profiling. Afterwards, an overview analysis over the content is done, in order to test the functionality and limitations of the prototype implementation and the proposed profiling approach.

Gathering

The processing was done with 8 threads and took 178 minutes for all 958,953 files. 316 files were not processed successfully, due to a bug in C3PO. The average processing time of 1000 files took 10.92 seconds.

Profile & Export

Generating the machine readable profile took 13.5 minutes. The resulting file was 700K in size. Repeating the profiling procedure with included object identifiers took nearly the same time and resulted in a file of almost 200 MB size. This rather large increase in file size shows how infeasible this strategy would be at even larger scales.

A problem that was revealed by the profile was that the data type of some of the characteristics were not correctly identified. This resulted in bogus aggregations for some of the numeric properties. Further investigation will be needed in order to determine whether this was caused by bad meta data or bad implementation in C3PO.

The export of the sparse matrix took about 2 minutes and 40 seconds and resulted in a 580MB file.

Analysis

Since there is no overview or ground truth data about the archive, we suspect to see that the web archive consists of mostly html documents, text documents (the stylesheets and scripts) as well as images. It is highly possible that there are a lot of conflicts, as the tools bundled in FITS often do not agree whether a file is xml, html or a mixture of both. Another assumption is that the average file size is around 0.2MB, which means that the overall size of the analysed web archive data should be a little less than 200GB.

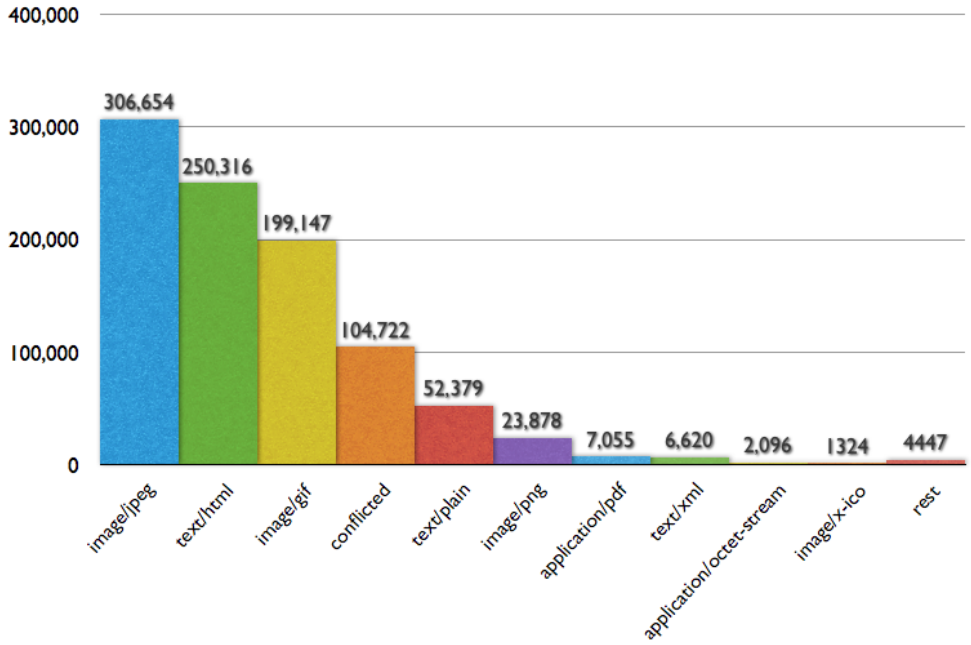

Figure 5.5: The most represented mime types identified within the web archive collection as C3PO showed.

C3PO revealed the following: In terms of size, the assumptions were wrong by a factor of almost 8. The analysed web archive data has an overall size of 26GB. The average file size is 0.03MB with a standard deviation of 0.66MB. The smallest file within the data was 1B and the largest 201.73MB. This proves how difficult it is to predict even the size of such a heterogeneous collection, not to mention other more complex characteristics.

Figure 5.5 gives a distribution of the mime types identified within the collection.

Hypertext Markup Language Observing this mime type distribution shows that the initial assumptions are mostly correct. At a first glance, it is curious that there are more images than HTML files. The reason for that could be the fact that the fourth most occurring mime type category is *conflicted*. Since such a high number of conflicts were expected due to known problems with FITS, we examine the conflicted subset first. As the formats were also *conflicted*, the only clue that C3PO was able to provide was the format versions. 50% had version '1.0'. Since

many formats could have this version, the PUIDs of the subset were examined. It revealed that all of these 'conflicted' files had one of the following: *fmt/101*, *fmt/102*, *fmt/96*. A quick check with the PRONOM registry showed that these are XML and HTML formats. From this two conclusions can be drawn. Firstly, it was fairly easy to identify the 'conflicted' documents by crossmatching the format version with other properties, such as the PUID. Secondly, the problems with the FITS conflict resolution for XML and HTML were shown. Another 30% of the same document subset had version *4.01* which is a HTML version. Looking at the raw meta data confirmed this assumption. If we sum up the correctly identified HTML documents and the ones that are *conflicted* but were proven to be html, then the initial assumption of having mostly HTML files will be correct.

C3PO provides a simple pre- and post processing mechanism that is used during the gathering phase, which might offer a solution to such problems caused by conflicted data. Writing a simple post processing rule that checks the conflicted values of the format property and overwriting them, if and only if they are XML and HTML before storing the meta data in this particular example. There are two problems with this solution. First, the current implementation does not allow dynamic processing rule binding, which implies that the rules have to be precompiled and poses an inconvenience to the user. Second, making use of such a rule is a rather dangerous decision and requires the knowledge of an expert who understands the data well and the implications in case of creating a wrong rule.

Concentrating on the correctly identified html subset (the second group in the mime type table) showed the following. The size of this subset was a bit more than 3GB. 94% of the documents were not valid opposed to less than 1% valid. The rest were unknown. Well-formedness revealed that 57% were not well-structured html documents, whereas 38% were well formed.

JPEG The 306654 JPEG objects amounted to 32% of the whole set. Their overall size was almost 8GB. Most of the images were formatted with version '1.01' (48%), closely followed by format version '1.02' (41%). The rest of the format versions were distributed around 10 other versions.

82% were valid and well formed. Almost 18% were unknown in terms of validity and well-formedness and the very small rest were either not valid, not well formed or both.

Also 82% were had the YCbCr colour space and nearly 18% had RGB. At first the authors thought that these are the same subsets as the ones in the validity and wellformedness experiments. However, a short analysis showed that this conclusion could not be drawn.

Once again, the creating date was not extracted out of many files. Most of the files (for which it was extracted) were created between 2003 and 2007.

Graphics Interchange Format The size of this subset was 1GB with an average file size of 0.01MB, which is usual for such web content. 97% of the GIF (animated) images were formatted with the newer enhanced version - 89a, whereas the rest had the older 87a version. Almost all of them were valid and well formed and a very small subset of objects (566) were both invalid and not well-formed.

As expected most of them were compressed with LZW[5], which is a lossless data compres-

[5] http://en.wikipedia.org/wiki/Lempel-Ziv-Welch

sion scheme and is common for GIF images. Nonetheless, there were 4 GIFs that had conflict in compression, which is rather strange. It would be interesting to look at the originals and check, why the conflict is reported. Unfortunately, the original content cannot be obtained in this case.

Again, only a few had associated creation meta data.

5.4 Observations

In this section we observe the outcomes of the experiments and try to give an objective evaluation of the C3PO prototype tool.

Altogether, the experiments showed, that C3PO could be rather useful in giving a complete overview of the identification data of rather large collections. In cases, such as the web archive data, where it is rather hard to get an idea what content there is, the identification data, such as mime types, formats and format versions, gives a rather valuable insight into the data (**R1**). Even in cases where there were conflicts, it was still possible to figure out the type of content by creating some filter conditions (**R5**). Considering that web archives, currently try to preserve web content by ensuring the bit streams are kept in tact and that access is still possible, the information that C3PO provides is rather useful.

The other use case showed that in cases of documents or images, where different preservation actions might be chosen, not only a good overview of the identification data is given, but also some specific information about some characteristics can be aggregated and displayed (**R1/R4**).

Nonetheless, there were two big issues regarding the meta data and thus the quality of C3PO. For one the data sparsity is a huge problem, as it often makes it hard to distinguish between format homogeneous subsets. What is more, the quality of the data is very important. Even if a measurement is reported, there is almost no way of knowing if this is the correct measurement, or a bug in the characterisation process.

Scalability was shown to be feasible for collections of size up to a million objects on a single commodity machine (**R1/R2**). Using better hardware will probably allow some vertical scalability but only to a certain point. The next logical step is taking advantage of the horizontal scalability capabilities of the back end, which still have to be tested. What is more, the gathering process can be also distributed on more nodes, in order to improve performance.

As far as the machine readable profile is concerned, it was shown that its creation is feasible(**R3**). However, its structure and usefulness has to be evaluated by integration with other software components. Two potential problems were shown by these experiments. First, including a long list of identifiers is a solution which will not scale due to the resulting file size. Second, it is questionable, whether or not the inclusion of all known properties in aggregated form is needed.

Export of the data in a sparse matrix was shown to be rather fast and helpful for the user (**R6**). However, this approach will most probably not scale for all data on larger collections, as spreadsheet tools will most likely have problems handling such big files. Nonetheless, exporting only subsets of the data can be rather helpful to a preservation expert. Another question to look at in future work is the use case scenarios of the matrix and consider adding new features to C3PO that can cover these use cases.

The current implementation allows to filter or correct some of the meta data that might be wrong during the processing phase. By implementing some pre-processing rules, it will be possible to enhance the quality of the data (e.g. XML/HTML identification). However, this task might lead to faulty results, as such assumptions should not be done lightly. Also the dynamic binding of such rules is not yet supported.

When looking at the selected samples, one potential pitfall was found. None of the proposed algorithms is able to deterministically select outliers in homogeneous sets, which is a hint for a further optimisation.

The current prototype implementation provides visualisations for each characteristic in form of histograms (**R4**). Nonetheless, it was shown that it will be helpful to combine some of the characteristics in other types of visualisations (e.g. scatterplots or bubble charts). In the case of GPS data or image resolution data, this could make a lot of sense.

Filtering showed that in a few easy steps it is possible to find some interesting facts about the data. Enhancing the filtering mechanism further, will be even more helpful to a planner. For example inverting the condition or adding a logical 'OR' to the conditions might be helpful.

CHAPTER 6

Summary & Outlook

6.1 Summary & Contribution

This chapter offers a brief summary of this work. It outlines the contributions of this thesis and discusses some open issues and future work.

Summary

With the rapid increase rates of digital data production, the problem of preserving digital content becomes more pressing than ever. All our personal and social information, cultural heritage and scientific findings are stored and managed in a digital form. Much of this content is born digital and there are no other copies.

Although there is an increasing awareness in research and business communities about the problem, there are still many people and organisations that do not act accordingly. Ignorance, misunderstanding of the problem or financial costs are just a few of the causes for digital disasters and data loss.

Digital Preservation tries to keep digital content findable, accessible, readable and understandable through time. And since preserving content is not a one-time single-step process, but rather an on going effort, there are many aspects that have to be considered. The community aims to find ways to preserve already existing content by making use of different strategies and tools, but also considers prevention as a valid strategy. In the future, so called preservation-ready systems should try to keep our content safe from the minute it is born, until it is no longer needed. The problem is that all software that produces or manipulates information and is used within an organisation or a preservation-worthy scenario should cope with these problems.

The current state of the art in preservation processes usually follows the OAIS model as a guideline and recommendation, due to the lack of a more specific framework for the domain. Although it has issues, it has been widely used and adopted by the community. One of its sub-processes is a decision making process that evaluates different preservation strategies, called preservation planning.

The process of preservation planning is currently highly dependent on manual input of experts, regarding the content that is preserved. Even though there is a high chance that the preservation planning process will never be fully automated, it could be greatly enhanced. Even with current technology, preservation planners can benefit a lot. Experts and stakeholders do not have an overview of the content they possess or manage due to the large volumes of data they have to deal with. As a result, high-level assertions and descriptions are usually used as the basis of the preservation planning process. Based on these and manually chosen random sample records, different preservation strategies and actions are evaluated. Even though this works in practice, it introduces two immense problems. Firstly, this approach does not scale, because of the manual fashion of gaining an overview over the content and the manual unstructured input that it provides. The creation of a complete executable plan can take up to weeks and often needs the attention of more than one person. Secondly, because of its unspecific nature it can lead to rather vague or at least biased results.

Another huge impediment for planners and preservation experts in the current state of the art is the selection of representative sample objects. This is a small subset of a larger collection that contains several objects considered representative to the whole collection.

Due to the volume of real-world collections, creating a more detailed overview and finding good representative samples that capture the essence of the collection is usually a hard, cumbersome and time-consuming task. Thus, planners usually rely on random selection or consider only one or two characteristics, such as the format and the size of the objects.

Contribution

This thesis contributes by proposing an approach of the content profiling process that can aid the preservation planning environment and ultimately digital preservation. As part of it, we also provide a prototype implementation of a tool that enables the creation of a content profile for collections of significant size in a reasonable time frame.

We define a content profile as a machine-readable aggregation of all important characteristics of a collection of digital objects. It includes relevant information, such as the size and volume of a collection, aggregated identification data, but also any other preservation related characteristic. The profile usually also contains a small set of representative samples and their identifiers within the source.

The approach consists of three simple steps that gather and process the characterisation data, aggregate it and provide interfaces for analysis and filtering.

By exposing an aggregation of the content containing, some significant data and overview of a collection of digital objects, other software components can make use of the data and provide feedback to a stakeholder in an automatic fashion. These may include violations of organisational policies, fluctuations in expected growth, unexpected formats, and more.

The prototype implemented as part of this thesis is divided into two logical components because of scalability issues. It consists of a command line application that processes meta data files and allows near data processing, and a web application for analysis. By separating the processing component off of the analysis component, the implemented framework provides better flexibility for integration in real-world scenarios and systems. For example, the command line can be integrated within a repository or even executed regularly (e.g. by a CRON job) on

the system near the data. This will allow the effective utilisation of resources in the usually more powerful backend. Through the a web application, a user can filter and analyse the content based on different criteria, browse the raw meta data, make use of different representative sample algorithms and export the data. The web application can be deployed in any (other) environment. It connects to the backend and makes use of its resources. By exposing a simple REST API, other components, can communicate with the tool and obtain a machine-readable content profile.

In the backend, a widely used document store is used that supports native map reduce jobs. These provide the needed capabilities for handling large amounts of data and provide results in feasible time spans.

The current implementation works well on collections of sizes up to million digital objects on one commodity machine. The backend could handle even more data on common hardware. However, the front-end seems to be the bottleneck in such cases. In order to overcome this hurdle, better caching mechanisms have to be implemented and some of the data transfer could be optimised. Nonetheless, the handling of collections of such sizes is still valuable to many organisations and practitioners.

The prototype was disseminated to the community during the 9th international conference on preserving digital objects [33] and through the Open Planets Foundation Blog[1], which generated quite a lot of interest. It was also presented on a special training event carried out by the SCAPE project. The responses of the community thus far have been rather positive and the tool starts to take up.

In the near future C3PO will be presented at developer events[2] as part of other EU funded digital preservation projects and will be further developed by the community.

6.2 Open Issues & Next Steps

The work discussed in this thesis can serve as the starting point for future research and development. It leaves space for optimisation but sets the foundation of a solid framework for generating machine understandable content profiles. The evaluation showed room for future enhancement. In the following we give an overview of some topics and questions that have to be considered in the future:

Starting with trivial things such as more visualisation types, in order to give an even better overview to the planner. Adding support for scatterplots and bubble charts, will allow the visualisation of two different characteristics, which will be rather helpful to a planner.

Filtering can be enhanced by providing more options, such as the negation of a condition and the logical OR operator. However, this can make the usage more complex and has to be considered carefully. During this optimisation, caching of some filter results has to be built into C3PO. This will make it faster and much more responsive. Other issues related to filtering is the exchange between different filters, without removing the old one.

One important part that is currently completely skipped, due to the focus of this work is user management. As content profiling deals with rather sensitive information, user management and support for user groups and anonymisation is very important.

[1] http://openplanetsfoundation.org/blogs/2012-11-19-C3PO-content-profiling-tool-preservation-analysis
[2] http://wiki.opf-labs.org/display/SPR/SPRUCE+Hackathon+Leeds%2C+Unified+Characterisation

Better and thorough integration with other tools will enable planners to do their work faster and more effectively. As part of this work and within the SCAPE project, C3PO was integrated with Plato - the planning tool and Scout - the monitoring component. The interface between these components is the exported machine-readable profile. Making the profile format more expressive and analysing which characteristics can be omitted and which can be presented in a more verbose fashion can enhance the integration.

Connecting C3PO with repositories will greatly benefit the whole preservation lifecycle, as the content profile will be generated directly within the repository interface. C3PO provides some interfaces for extension, which have to be validated and enhanced if necessary. Integration with repositories, such as RODA will be a great benefit to preservation experts.

By conducting special planning case studies on larger scale, the different sampling algorithms can be tested and evaluated for their effectiveness.

Considering the whole proposed concept of content profiling, there are a few places that have to be addressed in future work. For one, the removal and regeneration of a content profile has to be done in an efficient way, without the need of starting from scratch. Improving characterisation processes and providing ground truth data will most probably influence the quality of the extracted characterisation data. With increasing quality of data, also the quality of the profiling service will increase. In order to make sure that characterisation tools produce high quality data, much effort has to be spent on benchmarking processes and approaches. These will help to better understand the used tools and the data, which will help the content profiling process.

If future work addresses these problems and questions, content profiling will be greatly enhanced and thus planners will be able to make use of a more automated preservation environment and will be better supported throughout their preservation activities.

Bibliography

[1] Christoph Becker, Kresimir Duretec, Petar Petrov, Luis Faria, Miguel Fereira, and Jose Carlos Ramalho. Preservation watch: What to monitor and how. In *Proceedings of the 9th International Conference on Preservation of Digital Objects (IPRES 2012)*, Toronto, Canada, October 2012.

[2] Christoph Becker, Hannes Kulovits, Mark Guttenbrunner, Stephan Strodl, Andreas Rauber, and Hans Hofman. Systematic planning for digital preservation: evaluating potential strategies and building preservation plans. *IJDL*, 2009.

[3] Christoph Becker, Hannes Kulovits, Andreas Rauber, and Hans Hofman. Plato: a service oriented decision support system for preservation planning. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '08, pages 367–370, New York, NY, USA, 2008. ACM.

[4] Christoph Becker and Andreas Rauber. Decision criteria in digital preservation: What to measure and how. *Journal of the American Society for Information Science and Technology (JASIST)*, 62(6):1009–1028, June 2011.

[5] Christoph Becker and Andreas Rauber. Preservation decisions: terms and conditions apply. challenges, misperceptions and lessons learned in preservation planning. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, JCDL '11, pages 67–76, New York, NY, USA, 2011. ACM.

[6] Christoph Becker, Andreas Rauber, Volker Heydegger, Jan Schnasse, and Manfred Thaller. A generic xml language for characterising objects to support digital preservation. In *Proceedings of the 2008 ACM symposium on Applied computing*, SAC '08, pages 402–406, New York, NY, USA, 2008. ACM.

[7] Neil Beigrie and Maggie Jones. Digital preservation handbook. http://www.dpconline.org/advice/preservationhandbook, November 2008.

[8] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.*, 26(2):4:1–4:26, June 2008.

[9] The consultative committee for Space Data Systems. Reference model for an open archival information system (oais) : Recommendation for space data system standards: Ccsds 650.0-b-1. Technical report, The consultative committee for Space Data Systems, January 2002.

[10] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[11] Kresimir Duretec, Petar Petrov, Luis Faria, Christoph Becker, Rui Castro, and Miguel Fereira. Identification of triggers and preservation watch component architecture, subcomponents and data model. Public Deliverable D12.1, SCAPE, 2012.

[12] Luis Faria, Petar Petrov, Kresimir Duretec, Christoph Becker, Miguel Ferreira, and José Carlos Ramalho. Design and architecture of a novel preservation watch system. In *ICADL*, pages 168–178, 2012.

[13] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, Univeristy of California, Irvine, 2000.

[14] I. Foster, Yong Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE '08*, pages 1 –10, November 2008.

[15] Mykola Galushka, Philip Taylor, Wasif Gilani, John Thomson, Stephan Strodl, and Martin Alexander. Digital preservation of business processes with timbus architecture. In *Proceedings of the 9th International Conference on Preservation of Digital Objects (IPRES 2012)*, Toronto, Canada, October 2012.

[16] Mark Guttenbrunner, Christoph Becker, and Andreas Rauber. Evaluating strategies for the preservation of console video games. In *Proceedings of the Fifth international Conference on Preservation of Digital Objects (iPRES 2008)*, London, UK, September 2008.

[17] Helen. Heslop, Simon. Davis, Andrew. Wilson, and National Archives of Australia. *National Archives green paper [electronic resource] : an approach to the preservation of digital records / Helen Heslop, Simon Davis and Andrew Wilson*. NAA, [Canberra], 2002.

[18] Tony Hey, Stewart Tansley, and Kristin Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, Redmond, Washington, 2009.

[19] Steve Hitchcock and David Tarrant. Characterising and preserving digital repositories: File format profiles. *Ariadne*, (66), 2011.

[20] Matthew Hutchins. Testing software tools of potential interest for digital preservation activities at the national library of australia. *National Library of Australia Staff Papers*, 2012.

[21] Yannis Ioannidis, Seamus Ross, Hans Joerg Schek, and Heiko Schuldt. A reference model for digital library management systems. DELOS Research Activities 2005, July 2005.

[22] ISO. Open archival information system - reference model (iso 14721:2003). International Standards Organization, 2003.

[23] Andrew N. Jackson. Formats over time: Exploring uk web history. *CoRR*, abs/1210.1714, 2012.

[24] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, September 1999.

[25] Johan van der Knijff and Carl Wilson. Evaluation of characterisation tools. Technical report, Koninklijke Bibliotheek and Britsh Library, 2011.

[26] Hannes Kulovits, Andreas Rauber, Anna Kugler, Markus Brantl, Tobias Beinert, and Astrid Schoger. From tiff to jpeg 2000? preservation planning at the bavarian state library using a collection of digitized 16th century printings. *D-Lib Magazine*, 15(11/12), 2009.

[27] Kyong-Ho Lee, Oliver Slattery, Richang Lu, Xiao Tang, and Victor McCrary. The state of the art and practice in digital preservation. *Journal of Research of the National Institute of Standards and Technology*, 107(1):93–106, Jan-Feb 2002.

[28] Raymond A. Lorie. Long term preservation of digital information. In *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, JCDL '01, pages 346–352, New York, NY, USA, 2001. ACM.

[29] Feng Luan, Thomas Mestl, and Mads Nygård. Quality requirements of migration metadata in long-term digital preservation systems. In Salvador Sánchez-Alonso and Ioannis N. Athanasiadis, editors, *Metadata and Semantic Research*, volume 108 of *Communications in Computer and Information Science*, pages 172–182. Springer Berlin Heidelberg, 2010.

[30] Petros Maniatis, David S. H. Rosenthal, Mema Roussopoulos, Mary Baker, TJ Giuli, and Yanto Muliadi. Preserving peer replicas by rate-limited sampled voting. *SIGOPS Oper. Syst. Rev.*, 37(5):44–59, October 2003.

[31] Yannis Marketakis, Makis Tzanakis, and Yannis Tzitzikas. Prescan: towards automating the preservation of digital objects. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, MEDES '09, pages 60:404–60:411, New York, NY, USA, 2009. ACM.

[32] Feng Pan and Wei Wang. Finding representative set from massive data. Technical report, IEEE International Conference on Data Mining, 2005.

[33] Petar Petrov and Christoph Becker. Large-scale content profiling for preservation analysis. In *Proceedings of the 9th International Conference on Preservation of Digital Objects*, Toronto, Canada, October 1-5 2012.

[34] N. Press. *Understanding Metadata*. National Information Standards Organization Press, 2004.

[35] The Linux Information Project. Magic number definition, August 2006.

[36] Andreas Rauber and Hannes Kulovits. Digital preservation: Challenges, solutions, and approaches to accountable planning of digital preservation solution. In *Proceedings of the IST Africa 2009 Conference*, Entebbe, Uganda, May 6-8 2009.

[37] Andreas Rauber, Stephan Strodl, Carl Rauch, Hans Hoffman, Giuseppe Amato, Max Kaiser, and Heike Neuroth. Delos dpc testbed: A framework for documenting the behavior and functionality of digital objects and preservation strategies. DELOS Research Activities 2005, July 2005.

[38] V. Reich and D. S. H. Rosenthal. Lockss: A permanent web publishing and access system. *D-Lib Magazine*, 7(6), 2001.

[39] David S. H. Rosenthal. Format obsolescence: assessing the threat and the defenses. *Library Hi Tech*, 28(2):195–210, 1 January 2010.

[40] David S. H. Rosenthal, Thomas Robertson, Tom Lipkis, Vicky Reich, and Seth Morabito. Requirements for digital preservation systems: A bottom-up approach. *D-Lib Magazine*, 11(11), 2005.

[41] Jeff Rothenberg. Ensuring the Longevity of Digital Documents. *Scientific American*, 272(1):42–47, 1999.

[42] Rainer Schmidt. An architectural overview of the scape preservation platform. In *Proceedings of the 9th International Conference on Preservation of Digital Objects (IPRES 2012)*, Toronto, Canada, October 2012.

[43] Stephan Strodl, Christoph Becker, Robert Neumayer, and Andreas Rauber. How to choose a digital preservation strategy: Evaluating a preservation planning procedure. In *Proceedings of the 7th ACM IEEE Joint Conference on Digital Libraries (JCDL'07)*, pages 29–38, New York, NY, USA, June 18-23 2007. ACM Press.

[44] Stephan Strodl, Petar Petrov, and Andreas Rauber. Research on digital preservation within projects co-funded by the european union in the ict programme, May 2011.

[45] Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner. A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, 39(1):50–55, December 2008.

[46] Christian Weihs and Andreas Rauber. Simulating the effect of preservation actions on repository evolution. In *Proceedings of the 8th International Conference on Preservation of Digital Objects*, pages 62–69, Singapore, 2011. National Library Board Singapore, Nanyang Technical University Singapore.

[47] D. H. D. West. Updating mean and variance estimates: an improved method. *Commun. ACM*, 22(9):532–535, September 1979.

[48] J.M. Wing and J. Ockerbloom. Respectful type converters. *Software Engineering, IEEE Transactions on*, 26(7):579 –593, July 2000.