Document DNA: Distributed Content-Centered Provenance Data Tracking

A thesis
submitted in partial fulfillment
of the requirements for the degree
of
Doctor of Philosophy
in
Computer Science

at
The University of Waikato

by
Michael Rinck

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Department of Computer Science
Hamilton, New Zealand
July 29, 2015

Abstract

This thesis presents a new content-centered approach to provenance data tracking: the Document DNA.

Knowledge workers are overwhelmed as they find it hard to structure, maintain, and find re-used content within their digital workspace. This issue is aggravated by the growing amount of digital data knowledge workers need to maintain.

This thesis introduces a concept for tracing the evolution of text-based content across documents in the digital work space, without the need for a centralized tracking system. Our concept is inspired by the DNA common to life forms.

We present an analysis and comparison of research undertaken to support knowledge workers and review provenance data tracking systems. Provenance data has been used for data security, databases and to track knowledge workers' interactions with digital content. However, very little research is available on the usefulness of provenance data for knowledge workers. Furthermore, current provenance data research is based on central systems and tracks provenance at the file level.

We conducted three user studies to explore current issues knowledge workers face when working with digital content. The first study examined current knowledge workers' problems when re-using digital content. The second study examined to what extend the issues detected in our first study are addressed by document management systems. We found that document management systems do not fully address these issues, and that not

all knowledge workers make use of the document management system available to them. The third study examined reasons for low user saturation of available document management systems. As a result of these three studies we identified task categories and a variety of related issues.

Driven by these findings, we developed a conceptual model for Document DNA, which tracks the provenance of data used in the identified tasks. To show the effectiveness of our approach, we created a software prototype and conducted a realistic user study. Our software prototype is a Microsoft Word Add-In that tracks the evolution of content included in Microsoft Word documents. In our final user study, participants executed example tasks gathered from real knowledge workers with and without the support of our software prototype. The results of our study confirm that the Document DNA successfully addresses the issues identified. The participants were significantly faster when performing the tasks using the software prototype; most participants using traditional methods failed to identify the provenance of the data, whereas the majority of participants using the software prototype succeeded.

Acknowledgements

First of all, I would like to thank my chief supervisor, Annika Hinze. Without her encouragement and support, I would not have started this thesis, let alone finished it. I particularly admire her ability to put up with my weekly rants about everything. My other supervisors, David Bainbridge, Steve Jones, and Ryan Ko also deserve praise for providing insight and support when most needed.

Thank you to everyone in the Department of Computer Science, be it academic, technical, or administrative staff. Each of them contributed to this thesis in their own way, but always positively.

Profound thanks go to Anthea Robertson, Sam Sarjant, Craig Taube-Shock, Dave Snell, and Andreas Löf, for proofreading parts of my thesis. Without them, this thesis would not have been readable.

I would like to thank my friends and family for knowing when not to ask about my PhD and putting up with me when I was grumpy or miserable.

Finally I would like to thank Tine Ulrich, for sharing this journey with me and always putting things back into perspective. When my motivation was at the lowest, you were the reason that kept me going.

I gratefully acknowledge the financial support of the Department of Computer Science at the University of Waikato and the University of Waikato itself. Without that support, this thesis would not have been possible.

I would also like to sincerely thank everyone who participated in my user studies.

Contents

List of Figures

List of Tables

1

Introduction

It has been predicted that between 2012 and 2020, the amount of digital data in the United States will double every three years (Gantz and Reinsel, 2012). Keeping this data organized and accessible has become such an issue, that books have been published on how to navigate the *Data Smog* (Shenk, 1998). In addition, data has become increasingly mobile as it is now commonplace to collaborate using digital data. Recent security breaches (Black, 2013) add security and privacy concerns regarding the storage and access to digital data.

Knowledge workers primarily work with data and information that is both physical and digital. With the ever growing expanse of data that is born digitally, this means that they are even more affected than most by the issues outlined above since their work is centered on information, and therefore typically digital content. The amount of time knowledge workers spend maintaining and organizing their digital content is considerable (RealWire, 2013).

Digital content is often created by refining and combining already existing content, which again increases the mobility of the content. Research is a prime example of this, as we often build on previous knowledge to create new ideas and results. The information about the history of the creation process of content is often referred to as provenance data (Simmhan et al., 2005). Provenance data is metadata on content describing *who* changed the content *when*, *how* it was changed, and *what tools* were involved.

There has been ongoing research on metadata annotation systems to support knowledge workers. However, most of the proposed systems are file-centered and centralized, which does not address the high mobility of content. There is little research on provenance annotations and no research on whether or not such annotations would help knowledge workers.

This thesis investigates the use of distributed content-centered provenance data to support knowledge workers in coping with the increasing amounts of digital data. Specifically, it evaluates current and recent research conducted on metadata annotation systems, and includes three exploratory studies investigating current issues of knowledge workers and metadata annotation systems. The thesis proposes a model for distributed and content-centered provenance data tracking and a prototype implementing the concept, called Document DNA (DDNA). To evaluate DDNA, a user study was executed in which realistic data and scenarios from knowledge workers were used. The study measured the difference in result quality for tasks performed between users with access to provenance data and users without access. The study results show that users with access to the DDNA prototype produce task results of a significantly higher result quality than users who worked on the file-folder system.

1.1 Hypothesis and Research Questions

This section first outlines the hypothesis of this research. Based on the hypothesis, the research questions are developed, which motivated this research.

Hypothesis:

> Content-centered provenance data tracking increases the quality of the results of tasks knowledge workers perform when working with digital data.

Based on this hyptothesis, we now introduce the research questions guiding and directing this research.

1 What tasks do knowledge workers perform when working with digital content? Task in this context means every activity of a knowledge worker involving digital content. We need to answer this question to be able to

identify issues related to those tasks.

2 What are the issues with current systems aimed to support knowledge work? This question aims to assess the current situations knowledge workers face in their everyday work when using digital data. We need to assess what systems are available to knowledge workers and identify issues related to these systems. We can then determine if the identified issues can be addressed by a content-centered provenance data tracking system.

3 How can content-centered provenance data tracking be designed and implemented? If the answer to the second question includes that a provenance data tracking system would address the issues found, we need to know how to design and implement such a system. We need to target the tasks identified, and avoid the issues found, in order to be successful.

4 How can we measure the effect of using content-centered provenance data tracking? Measuring the result quality of tasks is a good way to detect if the issues found have been addressed sufficiently. A higher result quality means that issues are lessened or resolved. We need to determine how to measure result quality for the tasks we have identified to be able to evaluate the validity of the solution we propose. We need to find measurements that are both realistic and measurable in a controlled environment.

5 Does content-centered provenance data tracking increase the result quality of the tasks identified? Current studies tracking the provenance data generated by knowledge workers show that a high amount of provenance data is created by knowledge workers. However, there are no studies which objectively measure improvement gained on the knowledge workers' side when presented with that provenance data. The only evidence of improvement given is anecdotal (e.g., a knowledge worker told the researchers of insights gained from using their system). Therefore, we need to conduct our own study to evaluate if the information gained through content-centered provenance data tracking is increasing the result quality of tasks executed by knowledge workers.

The next section introduces the contributions made in this thesis.

1.2 Contributions

This thesis has seven main contributions:

1. A review and analysis of current and recent projects aimed to address the issues knowledge workers face when working with digital data, resulting in a requirements list for a more successful system.
2. An exploratory user study designed to discover issues knowledge workers have when working with digital data.
3. The results from a questionnaire that surveyed knowledge workers in businesses who use document management systems (DMS) (e.g., Microsoft Sharepoint) to explore issues these knowledge workers have in relation to those systems.
4. A case study examining issues with the introduction of DMS.
5. The design of a conceptual model for content-centered provenance data tracking for text-based digital documents.
6. The implementation of a distributed software system for content-centered provenance data tracking in Microsoft Word documents.
7. An evaluation of the usefulness of content-centered provenance data in a user study.

1.3 Thesis Structure

This section outlines how the chapters in this thesis address the research questions and what chapters include which contributions.

Chapter 2 This chapter defines the terms knowledge worker and provenance data in the context of this research. We also provide an overview of current issues discussed for knowledge workers using digital data, and identify task categories related to knowledge workers, to answer the first research question. The chapter also includes a discussion of metadata annotation systems that are either automated or user driven. We identify advantages and issues of the different systems and use these to create requirements for our own system, contributing to the answer of the third research question.

Chapter 3 The third chapter discusses provenance annotation systems that target knowledge workers. We discuss whether or not the systems address the requirements and use the insights gained to refine and affirm the requirements.

Chapter 4 This chapter answers the second research question by conducting three exploratory user studies investigating issues of knowledge workers using digital data. The first study is used to confirm issues identified in Chapter 2. To verify that these systems do not address the issues found sufficiently, the second study targets knowledge workers using DMS. The study also confirms the tasks identified in Chapter 2 and verifies that the DMSs do not sufficiently support them. The third study is a case study that exposes the issues connected to introducing a DMS to a work environment. The chapter finishes with a list of issues connected to the found tasks.

Chapter 5 The fifth chapter contributes to answer Research Question 3 by introducing the design for a distributed and content-centered provenance data tracking system, DDNA. The chapter finishes with a summary on how the requirements defined in Chapter 3 are met by the design.

Chapter 6 This chapter introduces our implementation of DDNA and is therefore answering Research Question 3. We include various examples and scenarios describing the functionality of the prototype.

Chapter 7 The final chapter answers Research Questions 4 and 5. We define the three attributes speed, accuracy and confidence to measure the result quality of knowledge workers conducting tasks. This chapter also includes the description of a realistic data set and realistic scenarios using this data set. Additionally, this chapter includes our evaluation study using the introduced scenarios. The study's results show a significant increase in result quality for the tasks performed in the given scenarios.

2

Background

In Chapter 1 we used the terms *knowledge worker* and *provenance data* to describe the motivation for this research. However, these two terms are broad and are used differently in different research fields. This chapter begins by defining what *knowledge worker* means in the context of this research. Also, the term *information* is clarified with regards to *knowledge workers*. We follow by identifying the issues of the file-folder system, currently used by most knowledge workers, including detailed examples. Afterward, we discuss automated and manual metadata annotation systems, because those aim to address the issues found with the file-folder system. Most of the metadata annotation systems annotate some form of provenance data, which is why we follow with a review of the definitions of provenance data in different fields. We also introduce our own definition of *provenance data*. This chapter closes with a summary of the requirements for a metadata annotation system based on the analysis from the discussed metadata annotation systems.

2.1 Knowledge Workers and Digital Data

What are *knowledge workers* and what differentiates them from other users of digital documents? Drucker (1964) was the first to use the term, but failed to provide a clear definition. Cortada (1998) ponders what constitutes a knowledge worker as follows:

What is a "knowledge worker?" What is "knowledge work?" How

> *are those questions different from "What is an Information Age Worker?" or simply, "What is the 'Age of Information'?" Is knowledge work tasks that require people to talk, write, think and apply knowledge? Doesn't a farmer talk, think and apply knowledge? What about a factory worker operating a CAD/CAM system driven by a computer with more capacity than all the computers build in the 1950s? Is a foreman on a construction job using a laptop to read blueprints and specifications not a knowledge worker? Or, are knowledge workers simply ministers, teachers, lawyers, accountants, librarians some managers, and consultants?*

This quote highlights how ambiguous the term is, despite being used for over 30 years. Originally the term knowledge worker was used to differentiate between people who mainly produce or maintain knowledge, and those who produce material goods. However, since the shift from the industrial age to the information age, the number and variety of knowledge workers have greatly increased. This is apparent in Cortada's statement, because more than half of his examples refer to computers and the use of digital information.

Reinhardt et al. (2011) attempted to clarify the term by introducing different types of knowledge workers, shown in Table 2.1. They define ten roles of knowledge workers and describe what actions they perform. Using these actions, we extract the following task categories that need supporting: *Information Search*, *Information Organization*, *(Co)-Authoring Information* and *Information Dissemination*. The other tasks are either assigned to less than four roles or do not require the use of digital documents. We will validate the choice of these tasks in Chapter 4.

With the general task categories established, we still have to clarify our understanding of tasks. The term task has been defined in many ways using very different approaches. Shepherd (2015) discusses a wide range of these approaches, two examples of which are defining tasks based on human behavior and based on the goals of activities. Our understanding of task is most similar to that of Papantoniou (2014), adapting and extending the definition, we define tasks as an activity with:

- an emphasized cognitive component (e.g. calculation, decision making)

- operating in a complex, changing environment (i.e. tasks cannot be predetermined), and
- involving digital content.

So far, we have used the term *information* without specifying if we mean digital, physical or both types. The concept of the *paperless office* may lead to a complete removal of physical information from the office, or at least remove the gap between digital and physical information, so we explore this concept first.

According to Sellen and Harper (2003), the establishment of the paperless office has been an ongoing goal of researchers and businesses since the mid 1970s. There are two main advantages of establishing the paperless office, firstly saving the need for paper, and secondly removing the need to bridge the gap between physical and digital information. However, the paperless office is far from being achieved in today's businesses. The U.S. Environmental Protection Agency estimates that each office worker consumes an average of 10,000 sheets of copy paper per year. The authors argued that paper will never completely vanish from office space and that the paperless office is a myth. They base their argument on statistics of paper usage growth (Sellen and Harper, 2003, p. 11) and the advantages paper holds over digital information. Sellen and Harper used the term 'affordances' to describe the advantages and disadvantages of different means to use information. Paper brings the following positive affordances into the work process of knowledge workers (Sellen and Harper, 2003, p. 17):

1. the ability to quickly share (printed) digital information without having to rely on digital means; and
2. the physical properties of paper allow for many human interactions (such as grasp and fold) that digital data does not support.

Since the introduction of tablets and smart phones has decreased the number of spaces into which digital information could not previously be brought and shared, the first point has become less relevant. However, the second point is still relevant, as lightweight flexible digital screens are not commonplace yet.

There have been efforts to bridge the gap between paper information and digital information stored in the digital space. Two recent examples

Role	Description	Typical Knowledge Actions (expected)
Controller	People who monitor the organizational performance based on raw information.	Analysis, dissemination, information organization, monitoring
Helper	People who transfer information to teach others, once they have passed a problem.	Authoring, analysis, dissemination, feedback, information search, learning, networking
Learner	People who use information and practices to improve personal skills and competence.	Acquisition, analysis, expert search, information search, learning, service search
Linker	People who associate and mash up information from different sources to generate new information.	Analysis, dissemination, information search, information organization, networking
Networker	People who create personal or project related connections with people involved in the same kind of work, to share information and support each other.	Analysis, dissemination, expert search, monitoring, networking, service search
Organizer	People who are involved in personal or organizational planning of activities, e.g., to-do lists and scheduling.	Analysis, information organization, monitoring, networking
Retriever	People who search and collect information on a given topic.	Acquisition, analysis, expert search, information search, information organization, monitoring
Sharer	People who disseminate information in a community.	Authoring, co-authoring, dissemination, networking
Solver	People who find or provide a way to deal with a problem.	Acquisition, analysis, dissemination, information search, learning, service search
Tracker	People who monitor and react to personal and organizational actions that may become problems.	Analysis, information search, monitoring, networking

Table 2.1: Types of Knowledge Workers according to Reinhardt et al. (2011),

are SOFIA by Jervis and Masoodian (2013) and the Human Centered Workplace (HCW) by Dighe and Hinze (2012). SOFIA allows for the linkage of physical folders and their contents in the office with digital counterparts, whilst the HCW allows for the linkage of printed documents to their digital versions, thereby merging the digital file-folder system with its physical counterpart.

While we acknowledge that physical information still has a part to play in today's office, this research focuses on digital information for the following reasons:

1. the amount of digital data increases faster than the amount of physical; and
2. there are means developed to bridge the gap between physical and digital data.

2.2 The File-Folder System — Issues

The digital file-folder system mirrors the paper-based files and folders that most offices use to store documents, where the folder and file names represent metadata of the files. However, we believe that the file-folder system is not ideally suited for handling large quantities of content and the flaws of this system are preserved within the digital file-folder system.

This observation that the file-folder system does not reflect the needs of the human mind was first made by Bush (1945). Barreau and Nardi (1995) were the first to conduct a study on user habits when organizing documents and when searching for documents stored on their PC. They found that people were content with the way the file-folder system worked. Digital folders usually serve as categories within which files are placed, and it is not common to create a large tree of categories to reflect files' attributes, such as time and place of creation for image files.

However, other researchers (such as Fertig et al. (1996a)) began to question the efficiency of the file-folder system, in particular the large amount of time spent on searching for files. Depending on the operating system and file type, additional metadata annotation options exist outside of the file-folder system. However, the file-folder system represents the constant

used for all files and is therefore our focus. We believe that the digital file-folder system is fundamentally flawed in the following ways:

1. It is file-centered, instead of content-centered.
2. It is inflexible, i.e., adding new categories to a file can result in the need to rework the existing folder tree.
3. Maintaining a file-folder system scales badly, because maintaining a large file-folder system is time consuming.

We now discuss these issues in detail using Example 1 as a base scenario, Example 2 for the issue of file-centricity and Example 3 for the issue of inflexibility and maintenance.

Example 1 *A user wants to store their papers according to whether or not they are long papers, short papers and whether or not they include pictures. The user also wants to store all illustrations separately. As shown in Figure 2.1, the user creates the folder* Papers *as a sub-folder of folder* Work. Papers *has 4 sub-folders,* Longpapers, Shortpapers, JournalArticles *and* Pictures. *The* Longpapers *folder also has a sub-folder for all files related to the CHI conference in 2013,* CHI'13.

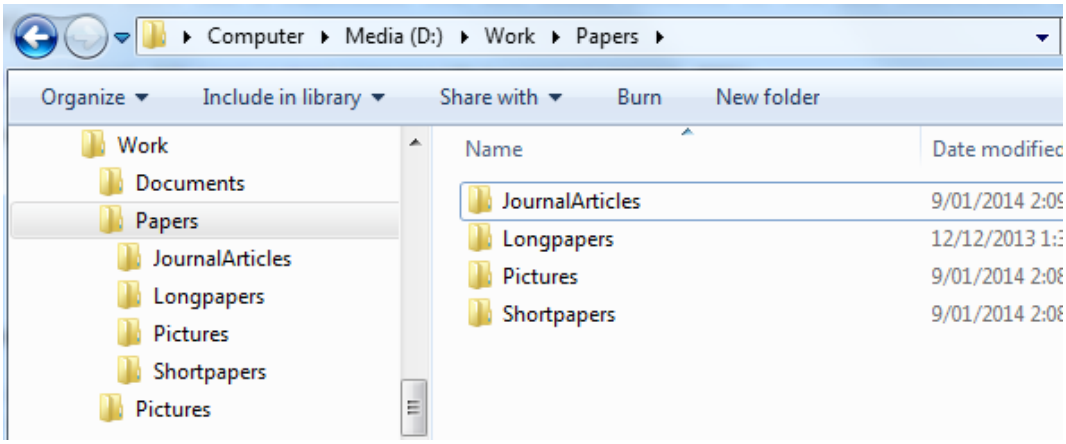


Figure 2.1: Example of a Folder with Files

2.2.1 File-Centricity

File-centricity in the context of this research means that a metadata system's focus is on the file level, instead of on the content level. This means that metadata specific only to parts of content in the file will be attached to the file. Following this principle, content in files will be associated with

metadata that is not necessarily related to that content, but related to other parts. Furthermore, the user is not able to distinguish which content piece was the trigger for a specific metadata annotation.

The main issue with file-centricity of the file-folder system is the limited scope of the metadata provided by it, and the increasing complexity involved with adding more metadata through sub-folders. In order to attach three pieces of metadata to a file, the user needs to create three folders. As shown in Example 2:

Example 2 *Following Example 1, the folder* /Work/Papers/Longpapers/Chi'13/ *includes several versions of a long paper written for CHI'13. However, the file* Chi'13_draft_with_figures.docx *includes* illustration_a *that is also used in short papers and journal articles. If the user decides that they want to be able to access all files including* illustration_a *at the same time, they need to create a folder* /Papers/Including_illustration_a/*, in order to represent all documents including a version of* illustration_a*, as shown in Figure 2.2. The file* Chi'13_draft_with_figures.docx *needs to be in two folders following that strategy.*

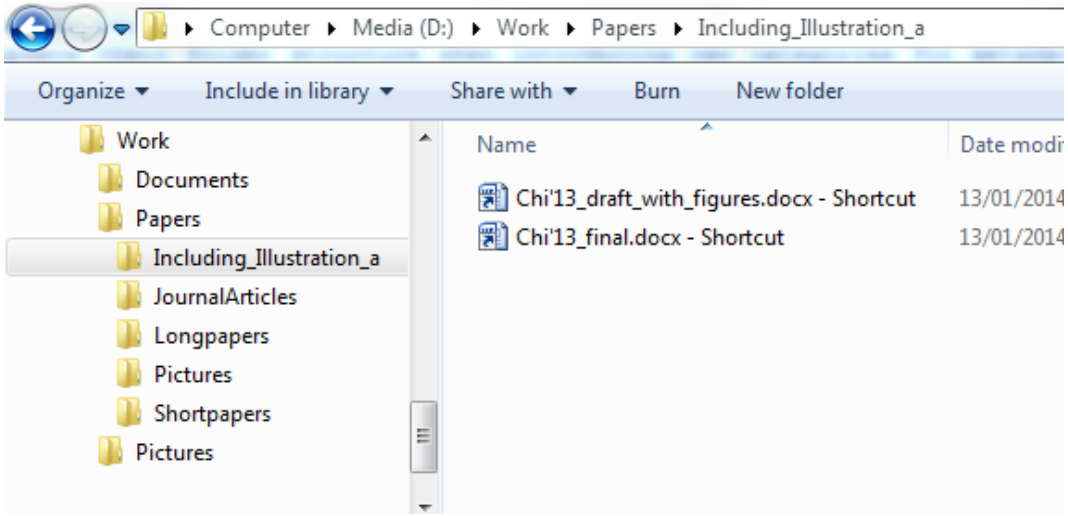


Figure 2.2: Changed Folder Structure

The file-folder system allows for each file to be linked in several folders, therefore allowing the user to attach multiple folder names (and even contradicting ones) as metadata to a file. For example, the file *Chi'13_draft.docx* could be linked in the short paper folder, whilst being stored in the long paper folder at the same time. However, if this option was used to the

fullest extent, it would need to ultimately represent all possible combinations of metadata.

2.2.2 Inflexibility

When users create new folders representing new metadata, they need to update their file-folder structure, as shown in Example 3:

Example 3 *If the user introduces two new folders* /Work/Papers/Accepted/ *and* /Work/Papers/Rejected/ *to the folder structure shown in Figure 2.1, they would need to update their existing folders by sorting the existing files and folders into one of the two new folders. The amount of time necessary for making such adaptation grows with the complexity of the existing folder structure, which often can be considerable. In this example, we need to sort files from three old folders into six new ones.*

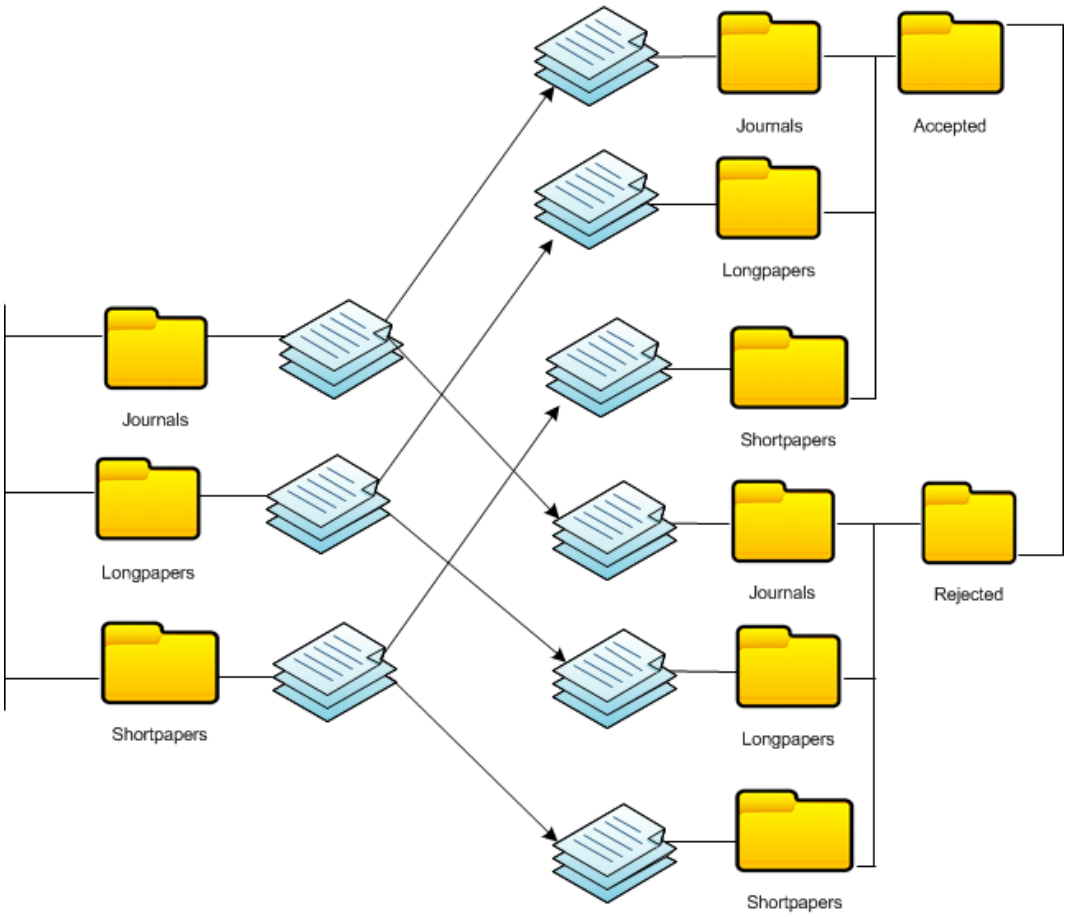


Figure 2.3: Sorting Files According to Changes Made to the Folder-Structure

Another issue is that even though a folder name represents metadata for content in a file, the two are separately stored. This separation means that when manipulating the content of a file, the change will not be automatically represented by the folder structure. Following Example 1, if the user decides to adjust the paper *Chi'13_final.docx* to be a short paper instead of a long paper, this change is not reflected by the folder structure automatically. Instead, the user needs to move the file into the new appropriate folder, which assumes that the user remembers to do so. This example shows that in order to keep the metadata annotation of a file-folder system correct, the user needs to maintain awareness of all locations in the file-folder structure where files are stored. The result is that users spend a large amount of time maintaining annotations when files are manipulated.

2.2.3 Scaling

There are two reasons why scalability is an issue associated with the file-folder system:

1. Linked Files / File Maintenance: The only way to overcome some of the file-centered limits of the metadata annotation in the file-folder structure is to make heavy use of the linking property of the system. This requires that every needed metadata tuple is represented by a unique folder path. The worst case complexity of the file-folder structure is exponential growth, because the possible metadata combinations grow exponentially. When updating content, users still need to be aware of all the folders to which they linked files to, because those links can become outdated due to file maintenance.

2. Folder Maintenance: The accuracy of the file-folder system relies on manual maintenance. Users need to be aware of all concerned files whenever they manipulate the folder structure, which in the worst case can be all files. The users may then need to re-arrange all files because of a change made to the folder structure.

These two points illustrate why the file-folder system is not scalable.

Over the last 15 years, several systems have been proposed to address issues with the file-folder system. One of the main approaches to replace the file-folder architecture has been Desktop Search. Desktop Search systems

rely on full text indexing and/or additional metadata (Chirita et al., 2005). Even though full text indexing has been added to many file-folder systems, it is not further discussed in this work, as the indexing also suffers from the same issues as other metadata annotation systems. These issues are listed at the and of the next section. Most metadata solutions rely on the addition of metadata annotation independent of the file-folder structure. While those systems address issues stated for the file-folder system, they have other issues that prevent them from replacing the file-folder system. Those issues will now be discussed.

2.3 Metadata Annotation — Manual and Automated

We divide metadata annotation into two categories: manual and automated. Manual metadata annotation systems are those systems that depend on the user to take action to create and maintain the metadata. Automated metadata annotation systems automatically create and maintain metadata. We introduce and discuss five manual and four automated metadata annotation systems to highlight general issues that prevent those systems from replacing or complementing the file-folder system.

2.3.1 Manual Annotation

This section introduces five systems supporting user-guided metadata annotation to files.

VennFS

VennFS is a new approach to display and sort files apart from a hierarchical folder structure by De Chiara et al. (2003). Instead of assigning files to folders, VennFS allows users to assign files to categories represented by elliptical shapes on a two dimensional plane. These shapes can overlap, so that a file can be assigned to several categories, as shown by the screen shot of the VennFS interface in Figure 2.4.

To assign files to different categories, the user can either drag the file into one of the elliptical shapes, or create a new elliptical shape around existing files. VennFS also allows the user to weight the strength of a category-file link. This is represented by the distance between files and the center of the

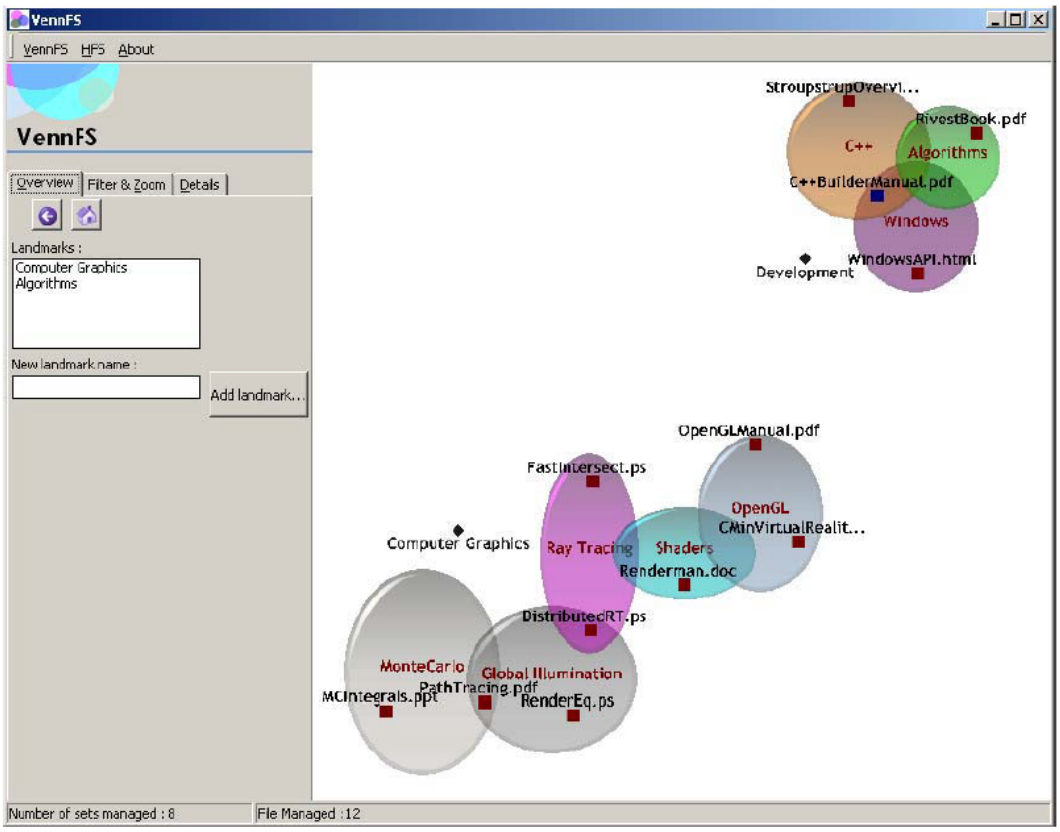


Figure 2.4: VennFS Interface as in De Chiara et al. (2003), ©2003 IEEE

shape. VennFS also displays time, such as the last point of access on a file automatically. This is done via color coding, red files are 'new' and blue files are 'old'.

The authors provide no details on how the assignment of files to categories is represented internally; we assume this is integrated in the VennFS file manager and not attached directly to the files through the file system functionality.

Analysis VennFS has the advantage of allowing for multiple categories for each file, and more importantly, for representing the strength of category file relationship. This augments the file navigation and search capabilities of users, because they can start with the strongest related files.

One disadvantage is the users' dependency on the VennFS system to use those relationships. Files that are transferred to other users will not include file metadata unless the other user is also using VennFS.

XanaWord Central Annotation System

XanaWord is an editing environment developed by Di Iorio and Vitali (2003), based on the Xanalogic concept by Nelson (1999). XanaWord's focus is the development and use of web pages. The authors stated that often users would re-use content of other web pages or include those sources in a new project. This re-use is accomplished by using links to re-used contents. Re-used contents may also be edited, leading to an update of the re-used content.

Since users do not always own the editing rights to content they want to re-use, XanaWord allows for users to edit the re-used content and store the new version separately in an online repository by storing the changes made. The changes are then applied dynamically whenever the resource is accessed. This results in a stored relation between the different versions, since the original source is always accessed when the new version is accessed.

Analysis One advantage of XanaWord is that it keeps track of the original source of used content. Users are able to find the original content that they adapted and re-used in their documents.

One disadvantage is the dependency on an SQL database. This means that users who are not connected to XanaWord cannot use the modified or original content, since the connection is not visible to them. DiIorio and Vitali did not discuss the details of the system's implementation.

Desktop+

Desktop+ is an approach that allows users to sort and access files independent from a hierarchical folder system (Fallin and Wyvill, 2003). Similar to VennFS, Fallin and Wyvill used a pile metaphor that is directly implemented on the desktop, instead of elliptical shapes in a separate environment. A pile represents a category. However, piles are not assigned any explicit identifier to represent the category the pile represents, forcing the user to memorize the categories. When a user moves the cursor over a pile, the pile opens up and shows all documents in an uncluttered way. The user is also able to apply queries directly to the Desktop+ environment to search for files. Whilst the user is in this search mode, a *radar view* is available, which still shows all the files. All this is visible in the screen

shot taken from the prototype in Figure 2.5. The authors also conducted extensive user studies to test the new system.

Analysis Desktop+ lacks the ability to represent the strength of a file-pile-relationship that VennFS has. Users are dependent on keeping their data in the Desktop+ system, since all metadata is stored in the Desktop+ system. This means the pile metaphor is lost outside the system. Desktop+ does not contain any advantage not already found in VennFS.

pStore

Xu et al. (2003) made a further step in the area of user-guided context annotation. They propose a system that does not support one schema (like piles or elliptical shapes), but instead introduces a framework called pStore that allows for the user to establish as many different relation schemes in between files as needed.

pStore builds on a flat file base (no folders) and only needs unique file or object identifiers. The architecture of the system is shown in Figure 2.6.

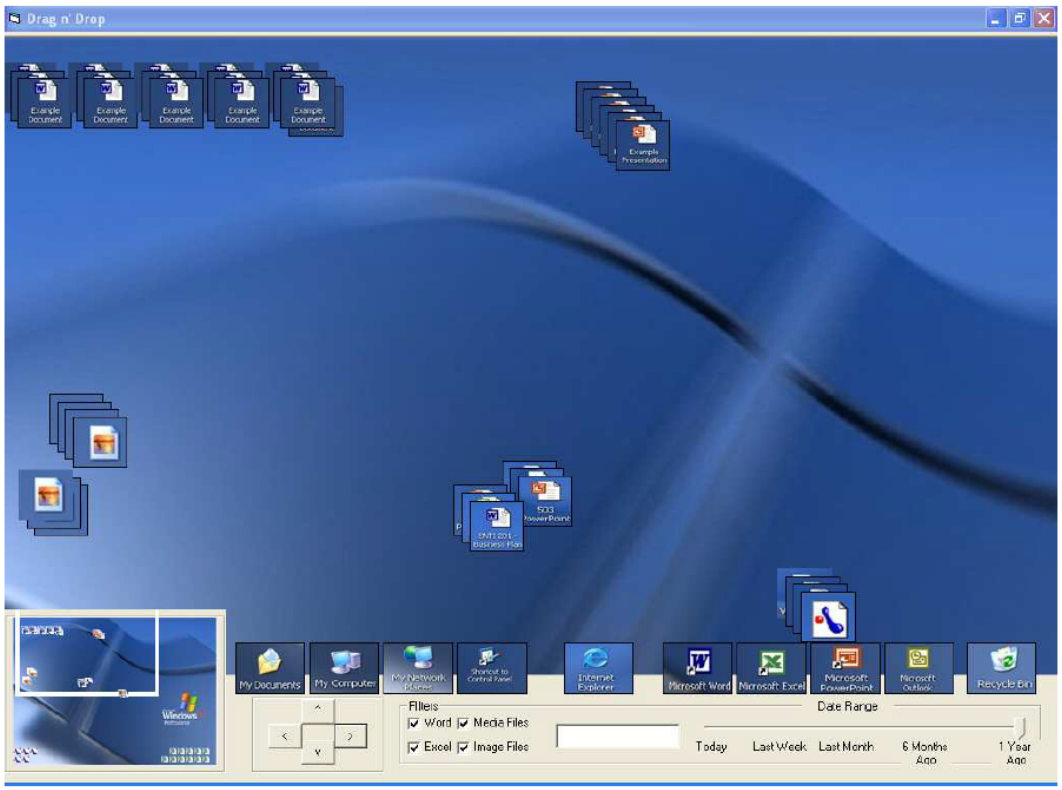


Figure 2.5: Desktop+ Interface Screenshot as in Fallin and Wyvill (2003)

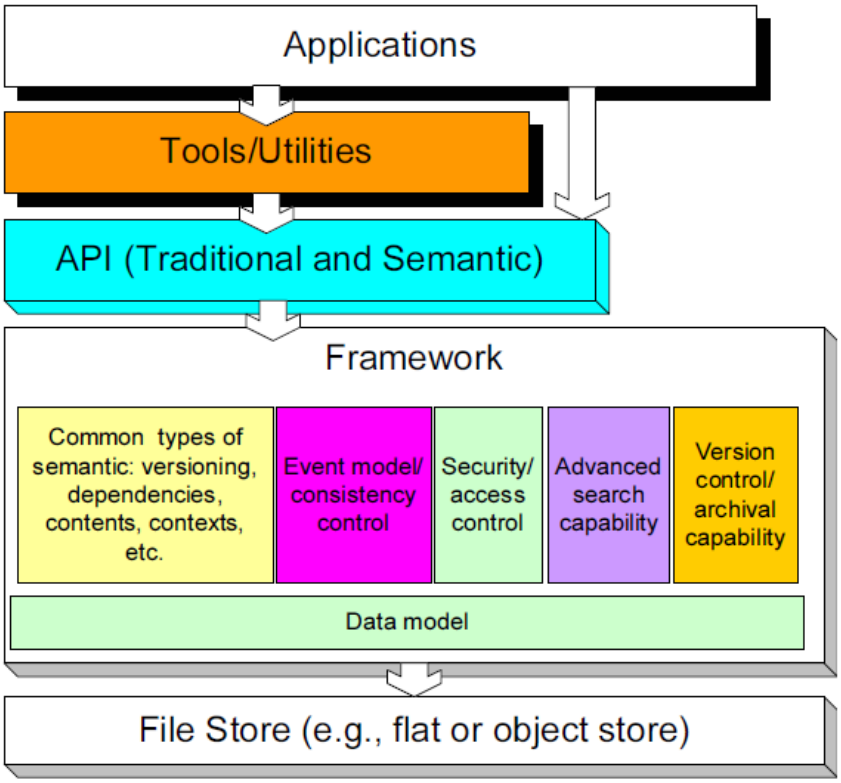


Figure 2.6: pStore Architecture as in Xu et al. (2003) Rights to individual papers remain with the author or the author's employer. Permission is granted for noncommercial reproduction of the work for educational or research purposes. This copyright notice must be included in the reproduced paper. USENIX acknowledges all trademarks herein.

The framework supports relations based on content, context, versioning dependencies and more. The relations are stored in RDF format apart from the files as subject relation object. Subjects and objects can both be files, whereas only objects can also be values. pStore introduces a valuable concept; new versions of files are always stored as new files and the connections to the old versions are kept as a relation.

Analysis pStore is advanced compared to VennFS or Desktop+ because of the range of describable relationships; however, it still has the disadvantage that files copied outside of the system will lose all the information about those relationships and that the user has to manually provide most relationships.

Archosum

Archosum is an organization software for files that aims to combine the benefits of keyword systems and a hierarchical folder structure (Hopkins and Vassileva, 2005). The authors' approach introduces the concept of an abstract entity. For example, Archosum links a file that is a 'song' to the abstract entity 'song', instead of linking all 'song' files directly to each other. A file can be linked to several abstract entities and can therefore be part of many categories. The authors mention a peer-to-peer approach to exchange abstract entities and the connections to the files, however, the links to the abstract entities are stored separately and are not usable without the Archosum system.

Analysis This approach is similar to VennFS and Desktop+, since it introduces the possibility to link files to more categories than a hierarchical folder system can. It shares the same disadvantage of the other two systems in requiring the user to assign files to their categories, which makes the system prone to errors or missing connections.

One advantage of the Archosum system is the possibility to exchange information about abstract entities with other users, thereby enabling the users to share the information about their files and the categories those files belong to.

We conclude that userguided context annotation systems do provide a variety of relationships and are very accurate, since the user provides the relations. However, we also found the reliance on both user input and a central repository to manage relations to be significant disadvantages.

2.3.2 Automated Metadata Annotation

This section introduces four approaches to automated metadata annotation. These systems use context as metadata. Context in this case refers to information associated with files, such as size or life span, or information about file use, such as any software that is used when the file is accessed.

Automated Context Annotation and Self-* Storage Systems

Soules and Ganger (2003) observe two approaches for *context annotation* for files in their paper, "Why can't I find my files? New methods for automat-

ing attribute assignment". These approaches are either based on content analysis or user input. However, they state that both approaches are unreliable because of their reliance on either user input (users make errors) or automated content analysis (automated processes are inaccurate).

They state that the context of files is much more likely to reveal meaningful relationships and therefore their approach tries to capture context. Soules and Ganger name two contexts that they find valuable: applications that were used when the files were accessed and file locations. They do not propose a solution for storing these contexts or how to display them to the user.

Mesnier et al. (2004) take an approach similar to Soules and Ganger (2003), whereby they automatically assign file categories based on their context. They are particularly interested in self managing systems, so they aim to create a storage system that is able to predict file contexts in order to store them accordingly. For example, a frequently accessed file needs to be stored on fast access storage.

Figure 2.7 shows the categories that Mesnier et al. aimed to predict and the policies they implement for those categories. Category prediction is based on decision trees and learning algorithms.

Analysis We consider the use of file context by Soules and Ganger to be a valuable approach and Mesnier et al. demonstrated that the approach can work for a storage management system. However, we believe that the contexts they chose were not the best suited for helping users organize their files because the technical values of files, such as size and lifespan, are not assumed to be that meaningful to users.

File class	Example policy
File size is zero	Allocate with directory
0 <size≤16KB	Use RAID1 for availability
File lifespan ≤ 1sec	Store in NVRAM
File is write-only	Store in an LFS partition
File is read-only	Aggressively replicate

Figure 2.7: Categories as in Mesnier et al. (2004) ©2004 IEEE

Documentation Know-how Sharing by Automatic Process Tracking

Prior to Soules and Ganger, Satoh and Okumura (1999) used file context for automated annotation. Their Know-how Sharing Agent tracks every copy action performed, as well as email and World Wide Web access that occurs while working with documents. This system is limited to the use of particular software, like Netscape or the Lotus Notes, which is dated by today's standards.

When a user creates a new document, the Know-how Sharing Agent identifies the keywords in the documents and finds other documents that use similar keywords. The Know-how Sharing Agent is able to provide sources and links that were used in the creation of the similar documents. Tracked data is stored in central databases and retrieved by the Know-how Sharing Agent when needed.

Analysis Satoh and Okumura considered the flow of information through a series of applications part of file context and consider this insight useful. Their system has the disadvantages of relying on central databases and specific software systems.

OmniStore

OmniStore is a storage and annotation system for personal area networks (PAN) proposed by Karypidis and Lalis (2006). The system provides a central storage backbone for data used in PANs, enabling the access and usage of greater amounts of data on small portable devices than the devices are capable of on their own. OmniStore also provides the user with one unified storage area for all devices connected to their PAN, freeing them of copying the needed data from device to device. Following the OmniStore architecture, every PAN based device uses an OmniStore Daemon to access the OmniStore repository.

Since access to files is always managed by the OmniStore repository, the system can take advantage of the available sensors of the accessing device, such as the GPS of a smart phone. This additional metadata can then be added to the repository, in the form of key value pairs. Karypidis and Lalis (2007) improved on their work, by allowing the key value pairs to be aggregated into more meaningful data; such as defining the context 'Hot Day' by a temperature range, and adding this context to every file

annotated with a temperature in that range.

Analysis The first advantage of the system is that the context is added automatically by the OmniStore backbone. This means that the user does not need to enter metadata into the system. The second advantage is that the system is connected to all devices in the PAN, meaning that the user can take advantage of the metadata with all devices that they use in the PAN.

A disadvantage is the need for a connection to the OmniStore repository. In case files are transferred to another user not using OmniStore, or not connected to that particular repository, the additional recorded metadata is unavailable.

Automated context annotation systems free the user from defining the relationships between entities. However, it also makes these systems prone to errors in detecting those relationships. The systems also fail to detect meaningful and important relationships when many relations are detected. Like user-guided annotation systems, the need for a central repository to manage the relationships is a disadvantage because the metadata use is limited by the reach of the system.

2.3.3 Summary of Metadata Annotation Systems

There are more systems available or proposed than we discussed here (Schütte, 1998; Fertig et al., 1996b; Bagga and Baldwin, 1998; Nelson, 1999; Schleimer et al., 2003; Grevstad, 2003; Ding et al., 2004; Boese and Howe, 2005; Signer, 2010). However, they do not add any new advantages or disadvantages we have not already discussed. We identified three issues in the reviewed annotation systems:

1. The annotations are stored separately, leading to the need of an additional centralized system to store and manage these annotations. Users who do not use the annotation software cannot access the annotations.

2. Either the user needs to create the annotations themselves or the annotations are created automatically. Both are prone to errors.

3. Annotations are stored in different formats and are not usable by other systems.

The third issue is supported by Svensson (2009), who stated that most annotation approaches limit annotations to context that can directly be gained from sensors, such as location, user or activity. However, each approach Svensson reviewed stored the contextual annotations in a different way, making it hard to re-use this information. We believe that this issue is magnified when using semantic annotations, since different approaches assign different identifiers to categories. It could be argued that cloud based systems, such as Google Docs, are slowly replacing the file-folder system. However, these systems also suffer from the above mentioned issues as they are based on file-folder structures.

However, there were two valuable concepts in the reviewed systems:

1. The concept of tracking the flow of information.
2. The concept of expressing different levels of strength of relations.

The flow of information named in the first concept is considered data provenance. Data provenance is a wide term and we therefore need to define it in relation to this work.

2.4 Provenance Data

Provenance of an object refers to how the current object's state came to be, e.g., the history of the object. Provenance for physical objects is typically limited to the history of the ownership and storage of the object. Provenance data is valuable for determining the authenticity and condition of objects like art, antiques, wine, books or collections of records. One example of such use is shown by Schibille et al. (2008), who discuss the provenance of late antique window glass from the Petra church in Jordan.

In this research, we limit provenance to information. For digital information provenance must be considered differently to that of physical objects. Both the storage and the ownership of digital objects are very different from physical objects. For example, physical objects cannot exist in several locations at the same time. The rate in which digital objects change location and ownership is also vastly increased when compared with the rate for

physical objects.

The other main difference between physical objects and digital objects is that physical objects are rarely manipulated in the same way, especially when goods of great value like antiques, wines or archeological artifacts are concerned. Digital objects on the other hand are constantly evolving. This is especially true when considering that changing location or ownership often comes with a change of format, which changes the underlying structure of digital information.

We now introduce the most prominent examples of provenance data of digital objects.

2.4.1 Provenance in Scientific Research

One of the cornerstones of scientific research is the production of data sets. The quality of produced data sets is often used to judge the overall quality of a piece of scientific work. Provenance is one of the tools used to measure the quality of a scientific data set. In this case, provenance means the methods, such as algorithms and experimental set ups used; and sources, such as data gained by sensors or cited sources, used to produce the data set (Barga et al., 2010). Provenance of scientific data sets is then used to judge the reproducibility and values of the data set.

2.4.2 Provenance in Databases

Data in databases is constituted of tuples. Using queries, these tuples can be manipulated, combined, aggregated and filtered in order to create views or result sets for data warehouses. According to Tannen (2008), the history of actions that preceded a view or result set is considered provenance for data sets in databases. They state that this provenance data is of high importance to establish the relationship between the result set or view and the source data used to create them. The nature of this relationship is of importance when judging the quality or suitability of the produced result set or view, because the relationship illustrates how the view was created. In databases, provenance is used to establish relationships between raw and processed data to give the users of this data a tool to judge the usefulness of the processed data.

2.4.3 Provenance in Semantic Web

The Semantic Web utilizes structured data (for example: ontologies) and automated reasoning, to allow users to answer complex questions using the world wide web as a knowledge source. Provenance in Semantic Web research is used to assess the quality of an answer gained using the Semantic Web, by including the reasoning process in addition to the sources used (Shadbolt et al., 2006). Issues often arise when the quality of the information sources are unknown due to frequent modifications, thus versions of sources and algorithms used are also important. In this domain, provenance is used to allow users to assess the quality of an answer gained through querying the Semantic Web.

2.4.4 Provenance as used in this Thesis

Our evaluation in Section 2.3.3 has shown that a focus digital files is not sufficient, instead an annotation system needs to recognize parts of files, which refer to as digital objects. Digital objects in this context are units of digital information that can be transported on their own and are meaningful to the sender or recipient. Provenance for digital objects is data that answers the following four questions, which we derived from the above – discussed definitions of provenance:

1. What is the origin of a digital object?
2. Which digital objects are derived from the current object?
3. Do digital objects share a common origin?
4. How do objects that share the same origin differ?

A digital object is considered to originate from another digital object if a series of manipulations lead from one object to the other.

This means that provenance in this context enables the user to find all other forms of an digital object (including origins and objects that originated from it) and provides the ability to determine which of those objects is the most recent one. These four questions can be encapsulated into the following two requirements:

R1 Relationship Detection — The system needs to be able to determine if two digital objects are related, i.e., is one originating from the other, or do they share a common point of origin? (Q1 & Q2 & Q3)

R2 Relationship Metric — The system needs to enable the user to determine the nature of the difference between two related digital objects. For example, how much do they differ in content? (Q4)

Additionally, we found that the reliance on a centralized architecture is a major disadvantage to an annotation system. The same holds true for relying on either manual user input for annotation, or inaccurate automatic annotation. We therefore formulate these two additional requirements:

R3 Distributed — The metadata needs be stored with the content, instead of separately. (Q3)

R4 Automated — The metadata needs to be created automatically and accurately. (Q3)

2.5 Summary

In this chapter, we introduced the terms 'provenance data' and 'knowledge worker.' We identified issues with the file-folder system and reviewed automated and manual metadata annotation systems that aim to address those issues. By analyzing the advantages and disadvantages of those systems, we derived four requirements for a successful annotation system. In the next chapter, we review five systems that track provenance data and discuss whether they meet these requirements.

3

Related Work

In this chapter we introduce five systems that utilize provenance data and discuss the advantages and disadvantages of those systems by analyzing whether or not they meet the requirements defined in the last chapter. We discuss the main disadvantage shared by these systems but also name insights gained from them.

3.1 Versioning Control Systems (VCS)

Versioning control systems, such as SVN and CVS (Apache Software Foundation, 2014; The CVS Team, 2008), allow users to keep track of all changes made to a set of files, called a repository. As shown in Figure 3.1, a central server hosts a repository which contains the data and a history of changes made to the data. Each set of changes between a save (commit) is called a diff. After the initial check out of a repository, a user can manipulate their local copy of the data included in the repository at will. After saving changes to the content, the user may opt to commit their new local copy to the central repository.

The server saves the new content, creates and saves a diff file containing the changes made or stores the old content as an old version. The current revision number is increased by one. A user can request updates to their local copy in case other users connected to the same repository have committed changes. Users can request to access older versions of the content from the central repository by specifying the revision number they desire.

There are decentralized VCS, such as Git (Hamano and Torvalds, 2014) and Mercurial (Mackall, 2014) that allow users to manage their repository locally and synchronize repositories with a common ancestor between users.

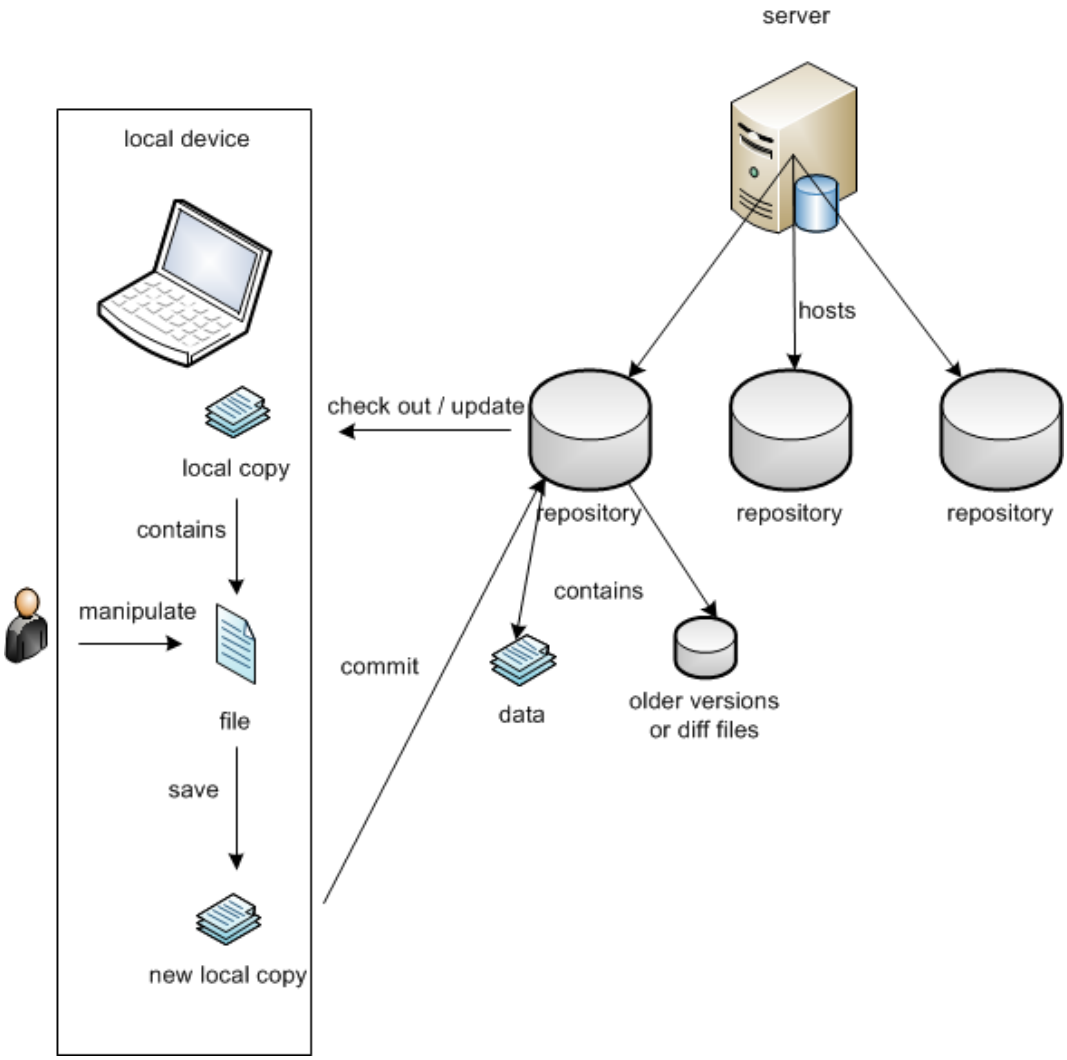


Figure 3.1: Work-flow for Versioning Systems

3.1.1 Analysis

We will now analyze VCS against the requirements listed in Section 2.4.4.

Requirement 1 The manipulation history of a file fulfills **Requirement 1** for different versions of the same file kept in a repository. However, copied and pasted content is not tracked, which means that **Requirement 1** is only partially fulfilled.

Requirement 2 This requirement can also only be partially fulfilled, as diffs between two versions of the same file can be used to determine the changes made from one version to the other but not determine if two files share copied and pasted content.

Requirement 3 This requirement is partially solved by distributed VCS. Users are able to create cloned repositories on their local machines and can therefore access and manipulate the included information at all times. At certain points the local repository is then pushed to the central repository or synchronized with other local repositories. If conflicts occur, such as other users committing conflicting changes earlier, merging algorithms are used. However, merging can be difficult and also result in loss of progress made on local repositories. The other versioning systems function completely centralized, therefore not meeting the requirement, since the user needs to keep all changes made in the repository. For example, a user might give a file to someone not using the versioning system and all manipulations they make would be lost to the central repository.

Requirement 4 Versioning systems require being set up for both the central system and the local users, as well as an amount of maintenance (for example: conflict resolution). This leads to users not using versioning systems for small projects as the amount of work needed to utilize a versioning system is to high in relation. However, small projects can grow and if a versioning system is set up later, much data may be lost already. We therefore conclude that Requirement 4 is only partially met.

An additional issue is one directly derived from the file-folder system, file-centricity. Versioning systems focus on files and the changes made in them. However, these systems do not track copied versions of the same content in other files, even if they are stored in the same repository. This can be vital provenance data and should not be omitted. A partial exception is made by GIT, which is able to detect copy/pasted content in a file through analysis of the diffs. However, GIT only allows to retrieve the original file name of the source, which becomes a broken link as soon as that file is missing or renamed.

3.2 Task Tracer

Task Tracer is a tool developed by Dragunov et al. (2005) to detect and store all actions performed by knowledge workers. Task Tracer aims to assign all detected actions to a task. The task needs to be specified by the user before they start working. Task Tracer implements a publish subscribe architecture, using available COM-Add-Ins in Microsoft Products, a CBT hook, the .NET FileSystemWatcher, a hook into the Windows Clipboard, and a hook into a phone modem to collect Caller-ID and speech-to-text information. Hooks are callback functions that are triggered when events defined in the hooks are detected, such as content copied into the clipboard.

Any detected action is stored as an event with a tag naming the task that was carried out at that time. The Task Tracer also stores the type of event, the time, the window handle of the active window at that time, the source of the event message (for example COM-Add-In in MS Word) and the version of the listener. Other tools can then subscribe to the Task Tracer to retrieve any events linked to a particular task, or all events.

3.2.1 Analysis

We will now analyze Task Tracer against the requirements listed in Section 2.4.4.

Requirement 1 This requirement is partially met, since the user can track tasks that involve the same files. Task Tracer also tracks copy and paste actions, therefore allowing users to determine if two files are sharing content that was directly copied and pasted from one to the other. However, since Task Tracer operates on file level, it cannot detect if multiple copy and paste actions involving several files means those files are related or not. For example, if a user copies one paragraph of text from document A to document B and then copies a picture from document B to C, the events supplied by Task Tracer could not be used to decide whether or not documents A and C are related or not.

Requirement 2 This requirement is met partially, as using the timestamps on save and save as actions allows for determining which file is the newer one when comparing two related files. However, it is not possible to make any statements about the semantic or syntactic difference between two re-

lated files, as the Task Tracer does not track inner-file content or the manipulations done to it.

Requirement 3 This requirement is not met, as Task Tracer centrally stores the collected provenance data. All actions performed outside of the scope of the Task Tracer are lost to the event data base and users are not able to retrieve any provenance data when not connected to the Task Tracer.

Requirement 4 This requirement is partially met. The Task Tracer requires the user to specify tasks when starting to work, so it is able to store the recorded events with the appropriate task tag. However, over time the Task Tracer is claimed to be able to adapt and detect tasks automatically.

The main contribution from Dragunov et al. (2005) in relation to this work is the realization that copy and paste are the most used actions in everyday work. Therefore, it is essential to track these actions and analyze the impact these actions have on content. The Task Tracer has been used to create a folder prediction interface (Bao et al., 2006), a website prediction interface (Lettkeman et al., 2006), a user task prediction algorithm (Shen et al., 2007) and a semantic search UI (Ghorashi and Jensen, 2013). It has also been used for text classification of emails (Keiser, 2009). While these projects do not address the underlying issues we found, they show that provenance data tracking is suitable for a wide range of tasks.

3.3 Versionset

Karlson et al. (2011) argue that one of the main problems in knowledge work is the rampant copying and versioning of files, which leads to a large cognitive overhead when keeping those files organized. The argument for rampant copying is supported by a study by Jensen et al. (2010). Versionset was also introduced by Karlson et al. (2011) to reconcile different versions of a single file into one entity. This was done to allow the user to quickly recognize if a file is the most recent version or if a newer version exists somewhere else.

The concept of this version set would also allow us to remove clutter from folders as insignificant files, in terms of version, could be faded out of view. Significant files are called milestones and are highlighted. However, Karlson et al. do not explain how the milestones would be differentiated

from less important versions of the files.

To create and maintain a version set, Karlson et al. deployed several tools to track save, save as, and copy and paste on users machines. Versionset then deduces the version history of a file by comparing the timestamps of the recorded actions and the involved copies of files. This information can be presented as a graph to the user. Karlson et al. conducted a study with a duration of several months. The study indicated that the information based on copy and paste actions is of great use to knowledge workers.

3.3.1 Analysis

We will now analyze Versionset against the requirements listed in Section 2.4.4.

Requirement 1 This requirement is partially met using Versionset. Versionset tracks similar events in comparison to Task Tracer and therefore allows for similar detection of relations between files. However, Versionset specifically focuses on files and their *versions*, therefore not processing interfile relations. Furthermore, this system falls short on tracking actions at the content level, making it impossible to decide whether or not a relationship chain including several files is meaningful.

Requirement 2 This requirement is partially met for versions of the same file as Versionset provides information used to determine the newest version of a file. However, no information is available regarding the semantic or syntactic differences at the content level. The requirement is not met in regards to copy/pasted content across files, as this was not the focus of the work.

Requirement 3 This requirement is not met as Versionset requires centralized tools to track and analyze the provenance data. The user will lose all provenance data as soon as they leave the reach of these tools.

Requirement 4 This requirement is met as Versionset functions without input from the user, i.e., versions of the same file are automatically detected and evaluated in comparison to other versions of the same file.

Karlson et al. contribute mainly by realizing that knowledge workers are spending much time on manual versioning of their files. Their approach

of addressing this through tracking of user actions involving these files is addressing this issue.

3.4 TrustCloud

Another motivation for tracking provenance is security and trust. Ko et al. (2011b) state that one of the key barriers for using clouds in businesses is the lack of trust that customers feel towards cloud computing. Ko et al. state that the massive amount of virtualization and data distribution within clouds results in the need for tools to track and secure the spread of sensitive end user data to enhance the trust users can place into the cloud.

For example, a user owning a file in the cloud should always be able to query who else has touched or modified that file and where else this file has been copied too. TrustCloud, developed by Ko et al. , is a concept designed to meet these needs. In particular, Flogger (Ko et al., 2011a) is the tool utilized in TrustCloud to track provenance data.

The architecture of Flogger consists of local Flogger listeners as well as Flogger listeners on virtual machines building the cloud. The listeners are recording actions processed at file system level, including the following non-exhaustive list:

- Virtual Machine (VM) File access date/time
- VM Accessed file name and full path such as /home/users/john/docs/sensitive.txt
- VM IP address
- VM MAC address
- Machine type i.e., VM or Physical Machine (PM)
- User Identifier (UID) & Group Identifier (GID) of file owner of the accessed file
- UID & GID of process owner who accessed the file
- Action done to accessed file, e.g., Create, Read, Write, Socket (Send Message), Socket (Receive Message), Delete

The data collected by the Flogger listeners is stored in local Flogs, which

get consolidated and stored into a central data store.

3.4.1 Analysis

We will now analyze Trustcloud against the requirements listed in Section 2.4.4.

Requirement 1 This requirement is partially fulfilled using Flogger, since it allows the user to determine if a file is the result of a set of manipulations (e.g., copy, save as) executed on another file. However, these actions are strictly tracked at the file level, so a copy action executed inside a file will not be detected. The ability to detect relationships is therefore very limited using the data collected by the Flogger system.

Requirement 2 This requirement is partially met. The user is able to determine which version of a file is the most recent one by analyzing the timestamps recorded by the Flogger system. However, this is the full extent of Flogger's capability to determine the nature of relationships between two files, and is therefore very limited.

Requirement 3 This requirement is not met, as Flogger depends completely on a centralized data store to consolidate provenance data collected by agents on local machines.

Requirement 4 This requirement is fully met, as Flogger functions are completely automatically after the initial setup.

Ko et al. (2011b) discuss an important use case of provenance data: trust and control. With the use of provenance data the user is able to track the spread of their data at all times, giving them trust and control in an environment they use. However, we believe that to completely fulfill the goal, the user must be also able to track content passed on between files, as this is a very common use case as shown earlier.

Provenance data has also been used to infer trust in relationships (Golbeck, 2006) and to create a peer to peer based web search that users can trust (Briggs and Smyth, 2008). However, these trust related approaches are beyond the scope of this research and are therefore not discussed any further.

3.5 Revision Provenance

Zhang and Jagadish (2013) acknowledge the need for provenance data when handling text documents, stating that the revision history of a document would supply the provenance data needed. However, Zhang and Jagadish also argue that a revision tree for a document can easily overwhelm the user with information when all they are interested in is a small part of the document and its history. For example, this is the case when several users collaborate on one document and a single user is only interested in seeing the changes made to a part of the document they are interested in.

The authors propose that in order to supply a user with provenance data for a particular piece of text, one needs to create a revision history centered on this piece of text and not the history of the whole document. To do this, the revision history of the document is transformed into several histories of revision units, which are formed by utilizing algorithms originally developed for database provenance by Zhang and Jagadish. The provenance of these revision units is then consolidated into one provenance data set and presented to the user. This method was tested on Wikipedia pages.

3.5.1 Analysis

We will now analyze Zhang and Jagadish's approach against the requirements listed in Section 2.4.4.

Requirements 1 & 2 Zhang and Jagadish's work is very similar to versioning control systems as it shares all the advantages and disadvantages regarding *Requirements 1 & 2*, since it utilizes the infrastructure of versioning control systems and the data collected by them. However, Zhang and Jagadish acknowledge the lack of accuracy caused by the file-centricity of versioning control systems and propose a method to detect and utilize a finer-grained unit of content to track provenance data. However, this unit is only tracked within one file's history, meaning the user still cannot track the provenance of data across files.

Requirements 3 & 4 Since the system relies on a preexisting revision history, it cannot be used decentralized, therefore failing *Requirement 4*. However, the analysis and creation of the revision units is fully automated, meeting *Requirement 3*.

3.6 Summary

In this chapter, we discussed four related systems for tracking and utilizing provenance data and versioning control systems in general. We defined the following four requirements in Section 2.4.4:

1. Relationship Detection — The system needs to be able to determine if two digital objects are related, i.e., is one originating from the other, or do they share a point of origin?
2. Relationship Metric — The system needs to enable the user to determine the nature of difference between two related digital objects. E.g., how much do they differ in content semantically and syntactically?
3. Distributed — The metadata needs be stored with the content, instead of separately.
4. Automated — The metadata needs to be created automatically and accurately.

An overview of the findings from this chapter is shown in Table 3.1. All of the systems have partial support for *Requirements 1 & 2*, but are limited to file level observations. The exception of this is by Zhang and Jagadish, who do track provenance on content level, but limited it to one file and its version history.

None of the evaluated systems were able to function fully without relying on a central source for data consolation and storage, which is a major disadvantage already observed in the metadata systems discussed earlier. Versionset and Trustcloud collect data automatically. Both supply tools to analyze and utilize the collected data.

	VCS	TaskTracer	Versionset	TrustCloud	Rev. Prov.
1 Detection	o	o	o	o	o
2 Metric	o	o	o	o	+
3 Distributed	o	–	–	–	–
4 Automated	o	o	+	+	o

Table 3.1: Fulfillment of Requirements in the Evaluated Systems:
– not fulfilled, o partially fulfilled, + fulfilled

All of the authors of the evaluated systems had some success when testing their systems in real world user environments and further studies ex-

ist supporting the usefulness of provenance data for knowledge workers (Jensen et al., 2010). However, we believe that the evaluated systems' main disadvantage is their disability to track provenance data at the content level instead of file level. We therefore conducted three exploratory studies to investigate this theory further.

4

Exploratory Studies

In the second and third chapters we confirmed that the current approaches are lacking as they are limited to file level tracking of data, which is insufficient. This chapter includes three user studies further investigating these issues and finishes with a conclusion discussing the implications we drew in regards to the design and implementation of our provenance data tracking system.

We first conducted an exploratory study with selected knowledge workers to test our assumption that users are more interested in provenance data that is applied and gathered at the content level. The study's goal was to address the first research question: What tasks do knowledge workers perform when working with digital content? The second study's goal was to confirm that the issues found in the related work are still present. Therefore the results are part of the answer to the second research question: What are the issues with current systems aimed to support knowledge work? This study is reported in Section 4.1 and is aimed at confirming the issues stated in Chapter 2.

In the first study, we found that most of the interviewed knowledge workers did not use a centralized document management system (DMS). To counter any bias this had, we therefore followed the first exploratory study with an online questionnaire aimed at 30 professional knowledge workers who have access to a DMS. We aimed to answer the second research question with this study, specifically targeting DMSs. The results of

the study are reported in Section 4.2.

One interesting side result from the second study was the fact that some of the targeted knowledge workers did not use, or know of, the DMS that was supplied for them by their employer. We conducted a case study within the work group of one of these knowledge workers to discover why they chose not to use the DMS. This study's aim was to answer part of the third research question: How can content-centered provenance data tracking be implemented?. The results of the study are reported in Section 4.3.

4.1 Exploratory Interview Study

The first user study was an interview series with knowledge workers that had at least five years experience of professionally working with digital content. This user study had two goals. The first goal was to explore what issues current users of digital documents have with the file-folder system. The results are used to verify if the issues identified in Chapter 2 are present in the everyday work flow of the participants. Secondly we wanted to verify the importance of relationships between content for the participants.

4.1.1 Study Design

The participants of this study have been chosen by contacting staff from the university as well as persons they suggested. For the guided interviews, we prepared a number of questions, but were also open to following up on anything interesting we would discover while talking to the participants. Statistical information such as age, gender and profession of the participants was also gathered. The interviews were conducted at the work places of the participants, to allow for quick access of their data. This setup would allow the participants to demonstrate how they managed their data. The interviews were structured around the following questions:

1. How many years have you been using digital documents in a professional environment?

2. What is your current most used document editor?

3. How often do you re-use digital content, on a scale from 1 (never) to 5 (very often)?
4. How often do you need to find re-used content, on a scale from 1 (never) to 5 (very often)?
5. How do you organize your documents?
6. What problems do you encounter in the organization of your documents?

We explained to each participant, that a digital document could be any digital file whose contents were accessed by the participant.

4.1.2 Results

We now give a summary of the results gained in the first exploratory user study.

Participants

We selected the participants from university staff and a local law firm. The only requirement was that the participants needed to have five or more years of experience handling digital documents. We aimed for a diverse group of participants and therefore selected participants from different areas of work. We had 20 participants overall: seven academics (computer science), four university staff members concerned with administrative or management tasks, three university staff members working at a library, four language teachers, one lawyer and one legal secretary from a local law firm. Participants were given the identifiers P1–P20. Eight participants were male and twelve female. One participant was between 20 and 29 years old, six were in the age group of 30–39, eight in the age group of 40–49 and five in the age group of 50–59 years.

Digital Documents and Viewers (Questions 1 and 2)

The participants had been using digital documents for five to thirty years (average 17.25). Most participants could not name a single editor they used most but rather named a range of editors they use every day. Figure 4.1 shows the results: Microsoft Word was the favored document viewer for

most participants. However, eight participants who mentioned Microsoft Word also mentioned another document viewer. These results also indicate that most document content is text.

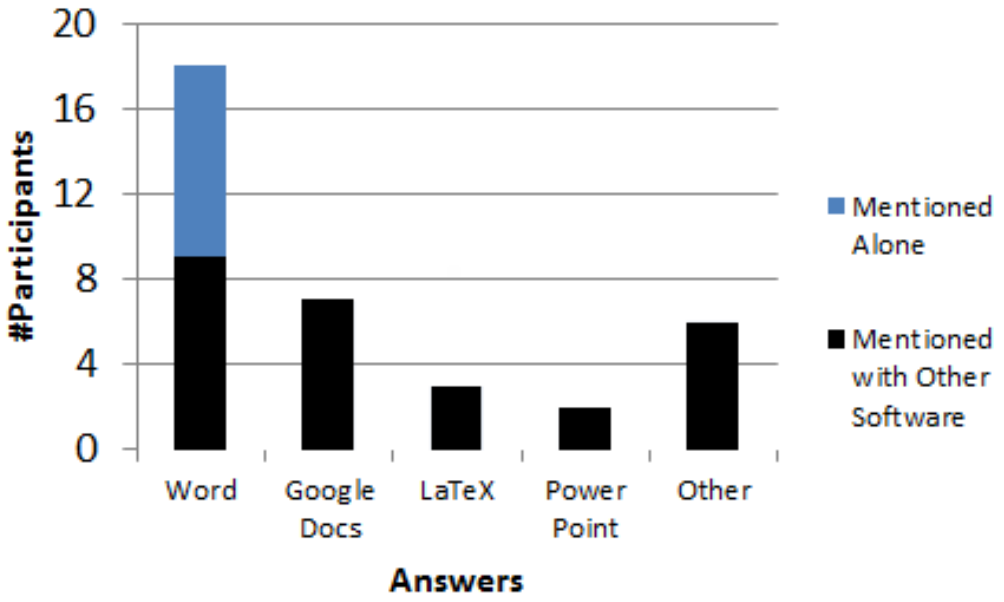


Figure 4.1: Document editors used by the 20 participants (multiple answers allowed)

Re-using Content (Question 3 and 4)

We explained that content was being re-used if it was taken by the participant from another digital source. No distinction was made between content that was changed when re-used, or content that was re-used in unchanged form. We also included the re-use of complete documents.

The participants' answers to Question 3 about re-using content were recorded on a Likert scale from one to five, where 1 meant 'never' and 5 meant 'very often' (see Figure 4.2 for results). Two participants chose a position in-between 3 and 4. We attributed one to 3 and one to 4. Every participants had re-used documents or document parts. The majority of participants re-used documents often and very often (4 and 5). The average of all the answers is 4 (often).

The answer to Question 4 for how often participants needed to find re-used content had greater variation (see Figure 3). We used a Likert scale to record the participants' answers. Five participants chose to answer in-between numbers (e.g., 3.5), so we split their vote for the figure (0.5 on 3

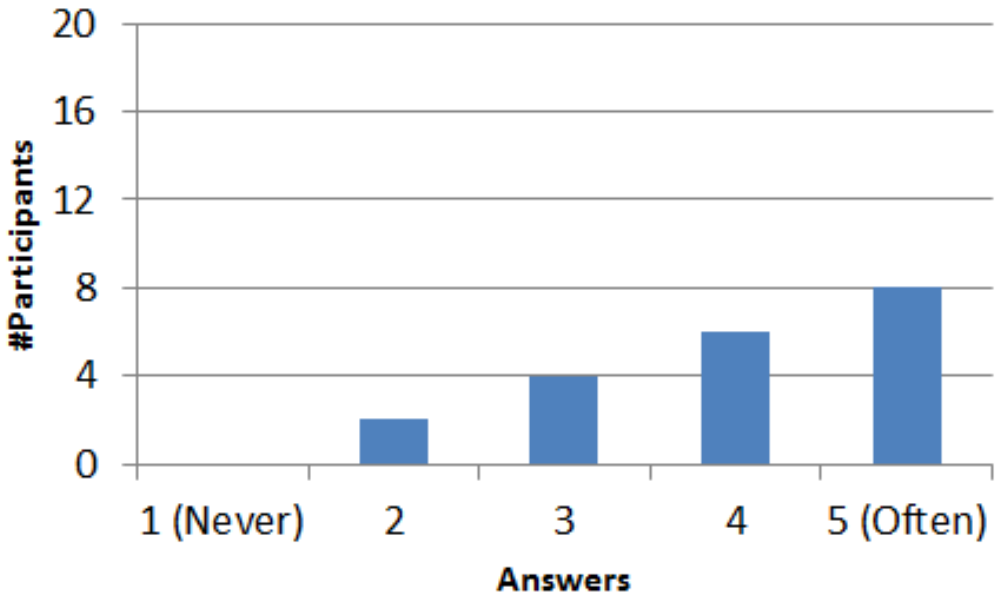


Figure 4.2: Frequency of content re-use

and 0.5 on 4). The average of all the answers is 3.225.

For some participants, whether or not they re-used content depended on the context of the content. For example, academics (P1, P3, P4 and P5) stated that they very rarely re-use or try to find re-used teaching content, but that the opposite was true for research content.

How Content is Organized (Question 5)

We made two main observations regarding the content organization habits of the participants.

File-Folder System The first observation was that the file-folder system was still used by all 20 participants to organize their documents. Each participant relied on some form of folder system to sort their documents: either the one supplied by the operating system, or a folder system supplied by another software, such as Dropbox or GoogleDrive.

The names of the files and folders were always very important to the participants. Participant 11 used the file names for exact versioning by incorporating dates, version numbers and purposes into the name.

The depth of the file-folder system used by the participants ranged from zero, meaning just the desktop, to 5 or more. Exact numbers were difficult

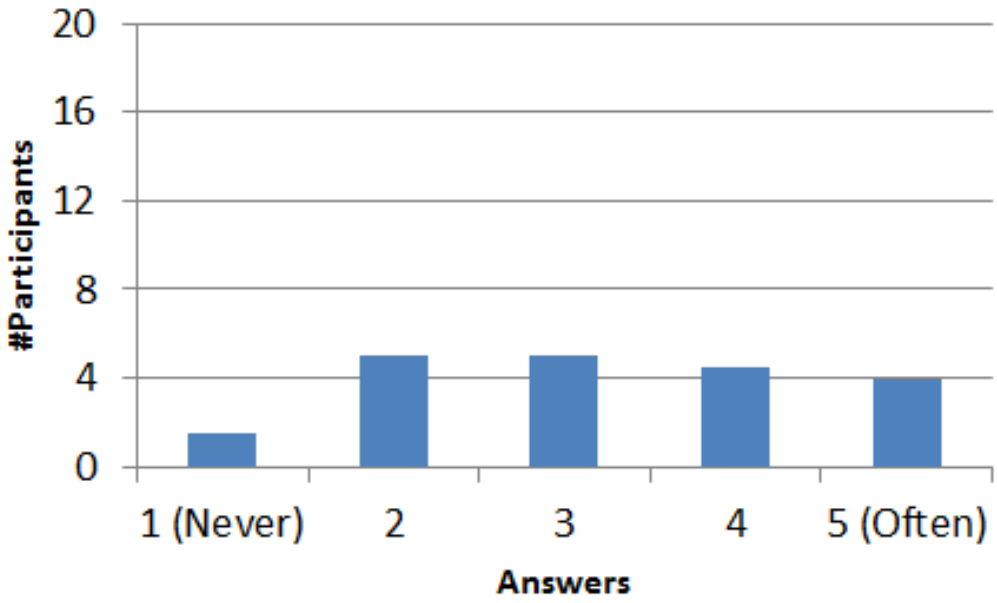


Figure 4.3: How often do participants want to find re-used content

to get in these instances, as users were usually not aware which of their self-created folders was the deepest. A separate document was kept to record locations of other documents in one of the work environments we interviewed participants (P14, P15 and P16) in.

Many Tools — Many Places The second observation we made was the large number of systems used to keep files and to organize them. Figure 4.4 shows the number of places in which the participants keep documents. Only P13 kept all their files in one place, and 11 of the 20 participants had three or more places where they would keep documents (P1, P2, P3, P4, P5, P7, P9, P10, P12, P15, P20).

Participants would often use different systems, depending on whom they collaborated with. For example, when P3 works with inhouse collaborators, they use Dropbox, but when working with outhouse collaborators, they use GoogleDocs. The pattern we observed was that participants who used many systems either had a heterogeneous (different software preferences) work group or many collaborators outside their work group.

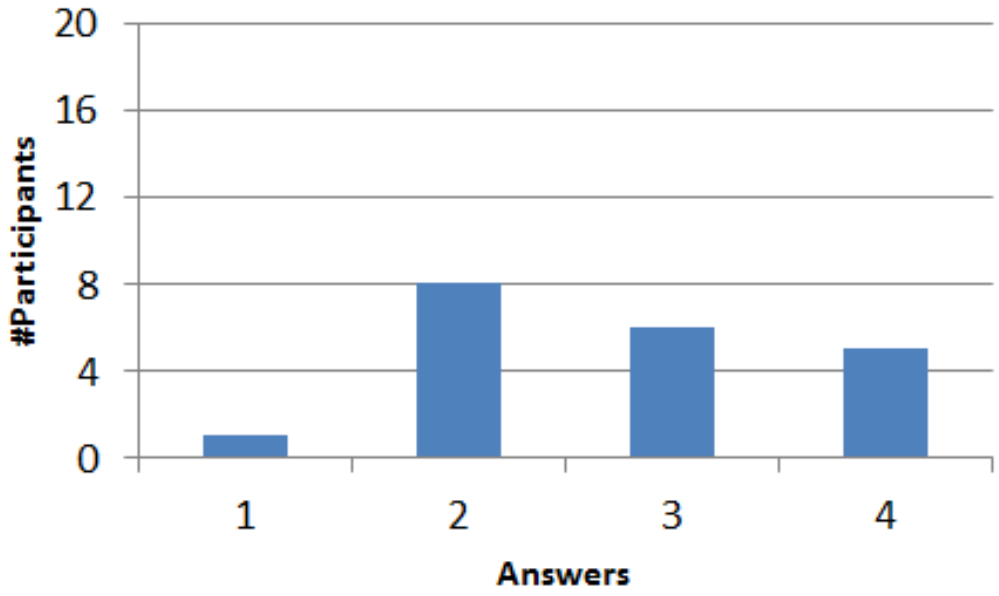


Figure 4.4: Number of places/computers in which digital documents are kept

Problems Encountered with Document Organization (Question 6)

Seventeen of the participants noted issues with the file-folder system (no issues: P6, P13 and P20). Ten participants stated that their file-folder system fails when they try to find documents they stored. The other seven participants named issues resulting from the limitations of the file-folder system, such as the lack of versioning or synchronization.

(a) Maintenance All but three participants (P6, P13 and P20) stated that the maintenance of their file-folder system required a considerable amount of work that they would rather spend on different tasks. This workload was reported to become unbearable as soon as participants interacted with people who had conflicting preferences as to how the file-folder system should be organized. Participant 14 also reported an instance where it was impossible to pass on the work of a retired co-worker due to the work being saved in the wrong folder system, which then became permanently locked after the retiree's account was deleted. Also, duplicates continued to be a problem for most of the participants who used a sophisticated folder system, even though they were aware of the options to link documents.

(b) Versioning Another issue mentioned by seven participants (P2, P12, P14, P15, P16, P18 and P19) was *versioning*. Dedicated versioning systems

such as SVN or Git were only mentioned and used by one participant from the area of computer science. Versioning issues were not necessarily connected to duplicates, but were also triggered by working in different environments on the same document. For example, the same document could be accessed via a home machine, a work machine, and a tablet PC. In general, versioning failed due to the attempt to manage it manually, instead of having a dedicated system. One participant in a management position stated: "Versioning is impossible to maintain in our work group, it does not exist." This was due to group members having different preferences for handling versioning.

(c) Synchronization Participants who used the same document in more than one system, or shared access to the same document with other users, mentioned the issue of synchronization (P1, P2, P3, P5, P8, P9, P11). They recognized the availability of tools for synchronizing documents, but were either not able to use them due to incompatibility between systems, or lacked the knowledge or time to set them up.

(d) Updating Instances of the Same Content There was one issue not directly related to the file-folder-system that was unique to the employees of the law firm (P19 and P20). They had a large body of legal documents that included many intentional near duplicates and lots of re-used content. Whenever a legal phrase changed, every document containing that legal phrase needed to be found and adapted according to the change that was made. This task was perceived to be very difficult. Because of this, lawyers were advised to check all the provided legal documents before using them, as they may have contained errors. Participant 20 also mentioned that letters from other law firms would frequently contain such errors, indicating that this issue is a widespread one.

4.1.3 Analysis

This study's aim was to provide answers to the first (What tasks?) and second (What issues?) research questions. No gender-specific differences were detected in the interviews, nor did we observe differences based on the age of the participants. The participants' years of experience with digital documents in a professional environment meant that the problems identified could not be explained by unfamiliarity with tools and systems.

Research Question 1

Re-using content and finding re-used content are tasks that were reported to be common for the participants, as shown in Figures 4.2 and 4.3. Additionally, the majority (19 of 20) of the participants have content in more than one location, as shown in Figure 4.4. These points imply that the focus of a new DMS should be on tracking re-used content inside the documents used by the knowledge workers. This is emphasized by the fact that participants referred to specific content snippets instead of documents when asked about re-finding content.

Research Question 2

We found that 17 of the 20 participants experienced problems while using the file-folder system, which can be considered the most common tool used by knowledge workers to organize their digital content. The most frequently mentioned issues were (a) maintenance of the file-folder system, (b) versioning of documents, (c) synchronizing documents, and (d) keeping track of instances of the same content. This confirms the issues with the file-folder system we identified in Chapter 2 and also confirms that these issues are answers to the second research question. This also confirms the need for a new approach to help users organize their documents.

Revisiting the Requirements

We defined the following requirements for our system in Section 2.4.4:

1. Relationship Detection — The system needs to be able to determine if two digital objects are related.
2. Relationship Metric — The system needs to enable the user to determine the nature of difference between two related digital objects.
3. Distributed — The metadata needs be stored with the content, instead of separately.
4. Automated — The metadata needs to be created automatically and accurately.

The issues (b), (c) and (d) are addressed by Requirements 1 and 2. The nature of relationships between content included in documents can be used directly to detect different versions of the same content, addressing issues

(b) and (c). These relationships can also be used to detect that two content pieces in different locations are the same piece of content, addressing issue (d). Requirements 3 and 4 address issue (a), since an automated and distributed system eliminates the need for user maintenance.

4.2 Usage of DMSs by Knowledge Workers

The first study confirmed the shortcomings of the file-folder system we named in Section 2.4.4. The results also imply that tracking re-used content across different physical locations is a major task for knowledge workers. However, we noticed that many participants in the first study lacked a DMS to support their efforts. Only two of the participants of the previous study (affiliated to the law firm) had access to a central DMS. We therefore conducted a second study targeting only knowledge workers who have access to a central DMS. In the second study, we aimed to answer two questions:

A Which tasks are the knowledge workers performing and are these supported by the DMS (to answer Research Question 1)?

B Do DMSs address the issues found in the first study (to answer Research Question 2)?

4.2.1 Study Design

To answer Questions A and B, we designed an online questionnaire targeted at knowledge workers with access to a DMS. The only requirements for participants were to be a knowledge worker and to have access to a DMS. The participants were contacted via email and encouraged to share the questionnaire link with colleagues in similar work environments. The questionnaire included the following questions:

1. Which of these tasks are part of your work process?
    - Analyze information
    - (Co-)Authoring information
    - Acquisition of information
    - Disseminate information

- Information search
- Information organization
- Learning
- Monitoring
- Networking
- Service search

2. With how many people do you collaborate using the same content?
3. Do you use a document management system? (If so, which?)
4. If using a document management system, how often to you use it for collaboration with colleagues?
5. Do you use a versioning system?
6. How often do you work with documents that are not stored inside your document management system?
7. Does your content management system support search of re-used content?
8. If no, would you like the system to support finding re-used content?
9. For which of your work processes are you utilizing the document management system?

We recruited participants via company contacts we had, which included companies in New Zealand, Germany, USA and Singapore. The third question asks for the general usage of a DMS, because we had no guarantee that the participants were actively using the provided DMS. To provide anonymity, participants are identified with P1–P31.

Question A Question 1 was used to define the tasks the participants mainly perform, so we can determine if tasks are linked to specific issues. Question 2 was used to discover if the tasks performed by the knowledge worker are performed in a group setting or alone. These two questions are aimed at answering the first part of Question A. Questions 3 and 4 are used to determine if the participants are utilizing the DMS they have access to and whether or not the document management helps with the participants'

tasks. Questions 3 and 4 therefore answer the second part of Question A.

Question B Question 5 was used to discover if a versioning system is used. This question was asked since one of the main issues found in the previous study was versioning of files. We wanted to discover if the participants with access to a DMS would have access to versioning and whether it is the DMS itself or an additional system. Question 6 was used to discover how much of a participant's content is handled by the DMS. This question is used to answer Question B, as some of the main issues found in the first study were synchronizing documents and keeping track of instances of the same content, which is a problem when content is handled inside and outside of the DMS. Questions 7 and 8 were used to determine the DMS's support for re-used content, as the previous study had shown many issues related to the re-use of content. Question 9 was used to determine which tasks the participants seek to support with the use of a DMS.

4.2.2 Results

We now present our results, beginning with the statistics regarding the participants and following the questionnaire.

Participants

The study had 31 participants, 23 male and 8 female. Figure 4.5 shows the age distribution, the participants were of the ages 18–25 (4), 26–39 (16), 40–59 (10) and 59+ (1).

Of the 31 participants questioned, 19 used Microsoft Word, 2 used Open Office and 9 used another content editor, as shown in Figure 4.6. Multiple mentions were not allowed this time as we were only interested in the main editor.

Question 1 and 2

Figure 4.7 shows the tasks that the participants stated were essential to their work processes. The only tasks that were not named 20 times or more were: Information organization (19), Monitoring (15), Networking (12) and Service search (6).

Question 2 asked how many people the participants are collaborating

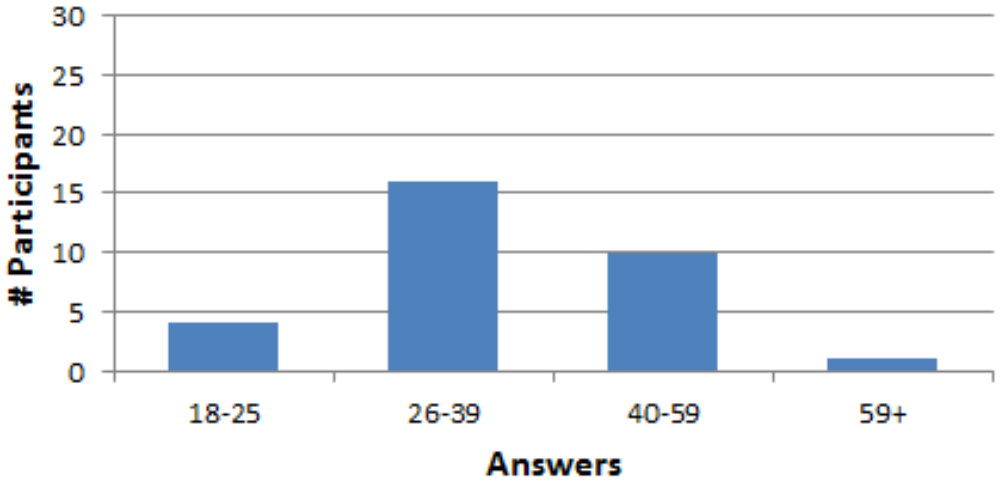


Figure 4.5: Age Distribution

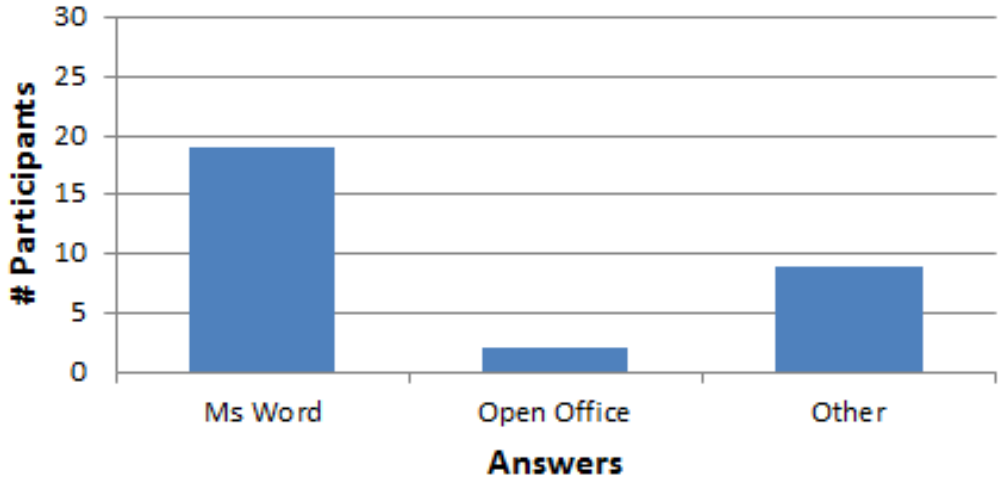


Figure 4.6: Document Editors Used by the 31 Participants

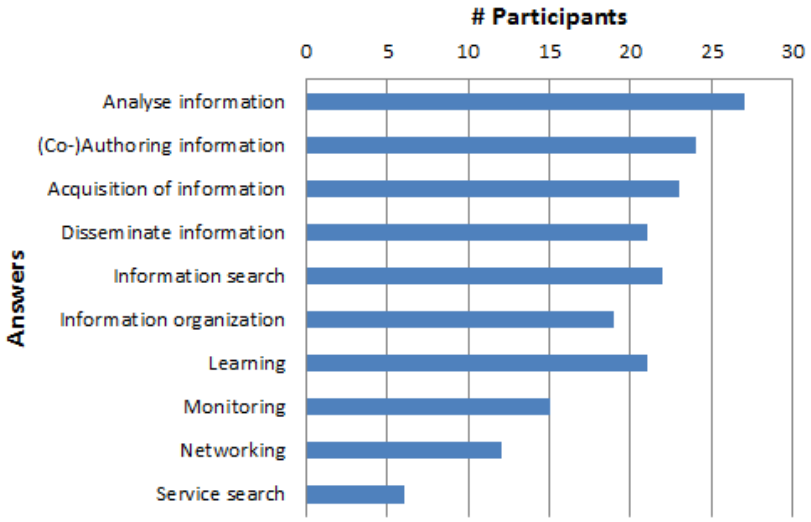


Figure 4.7: Which of these tasks are part of your work process?

with on the same content: No one answered 'just me', 6 participants answered 'one or two', 7 participants answered 'three to five' and 18 participants answered 'six or more', as shown in Figure 4.8. This means that every participant is collaborating with someone and more than eighty percent of the participants collaborated with three or more people.

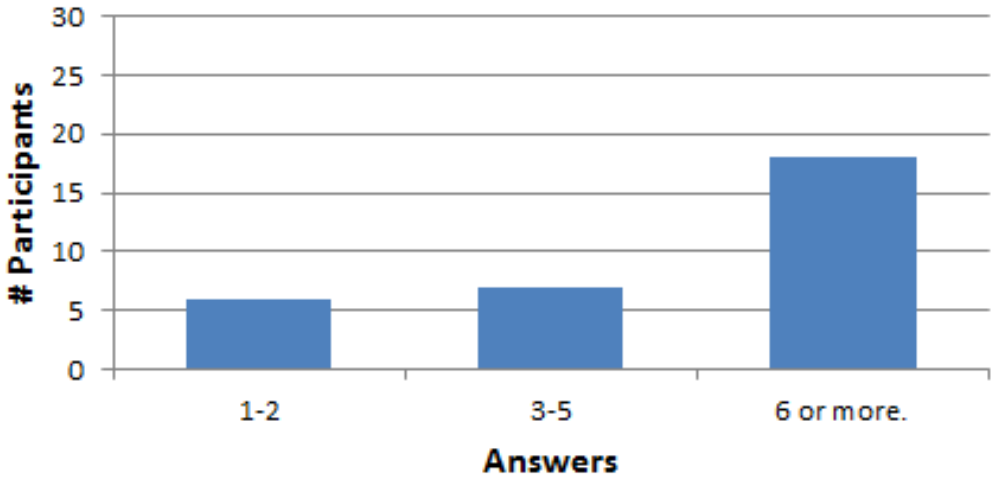


Figure 4.8: With how many people do you collaborate using the same content?

Question 3 and 4

Figure 4.9 shows how many of the participants use a DMS and if so, which system. Five participants answered that they do not use a DMS, seven

participants use Microsoft Sharepoint and 19 use some other DMS. Each participant had access to a DMS.

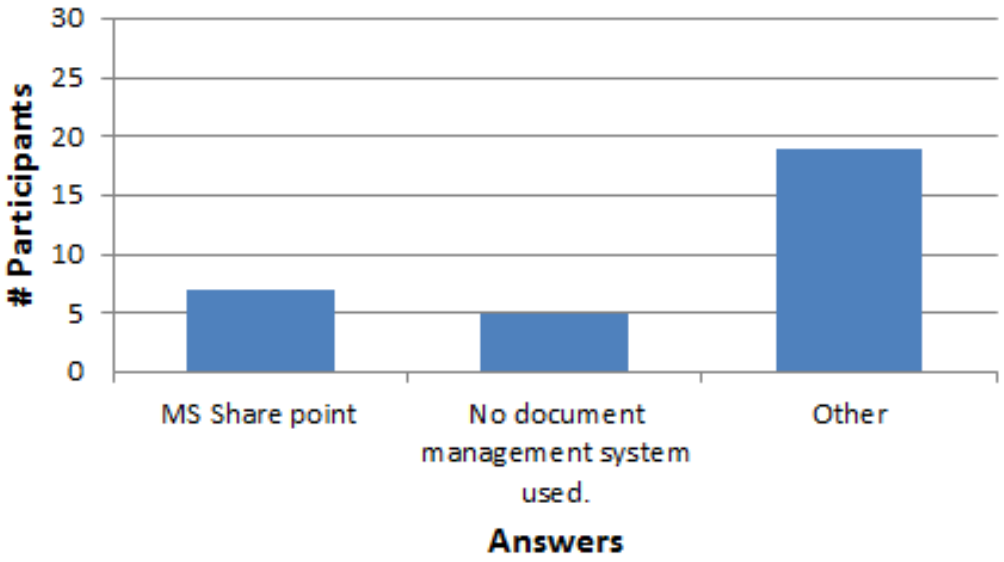


Figure 4.9: Do you use a document management system? (If so, which one?)

We asked participants how often they use a DMS when collaborating and the results are shown in Figure 4.10. Seven participants answered 1%–25%, three participants answered 26%–50%, five answered 51%–75% and twelve participants answered 76%–100% of the time. No participant was unsure (do not know) about this question. Only five participants claimed to never use the DMS for collaboration, but two of those participants answered that they don't use a DMS at all. The other three participants that claimed to not use a DMS answered this question with: 1%–25% (1), 26%–50% (1) and 51%–75% (1). We assume that those participants misunderstood the question, meaning 'How often do you collaborate using a document management system', as they might receive content through a DMS without actively using it for themselves.

Questions 5 and 6

Figure 4.11 shows that of the 31 participants, 26 use versioning. Of the other five participants: one did not know if versioning was used; one did not want to use versioning; and three had no software support for versioning.

We asked how often participants worked with documents that were not

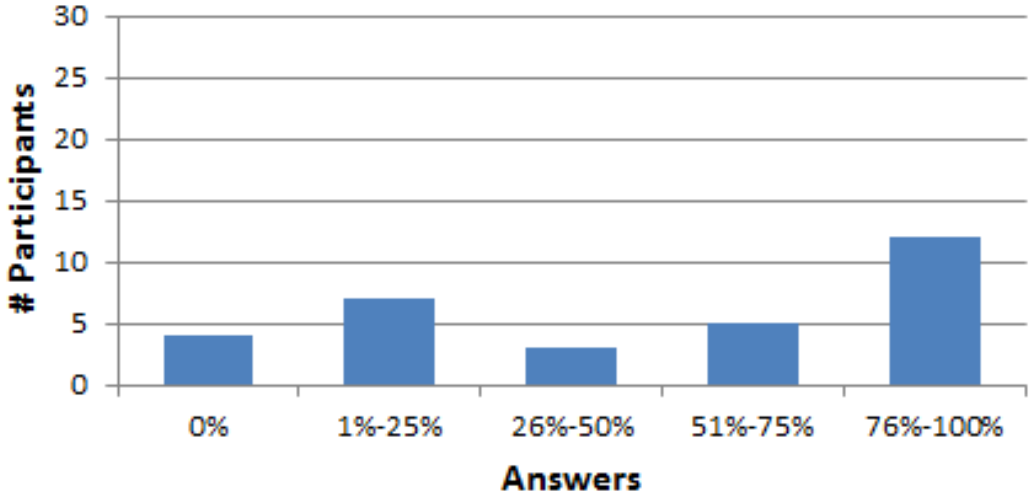


Figure 4.10: If using a document management system, how often do you use it for collaboration with colleagues?

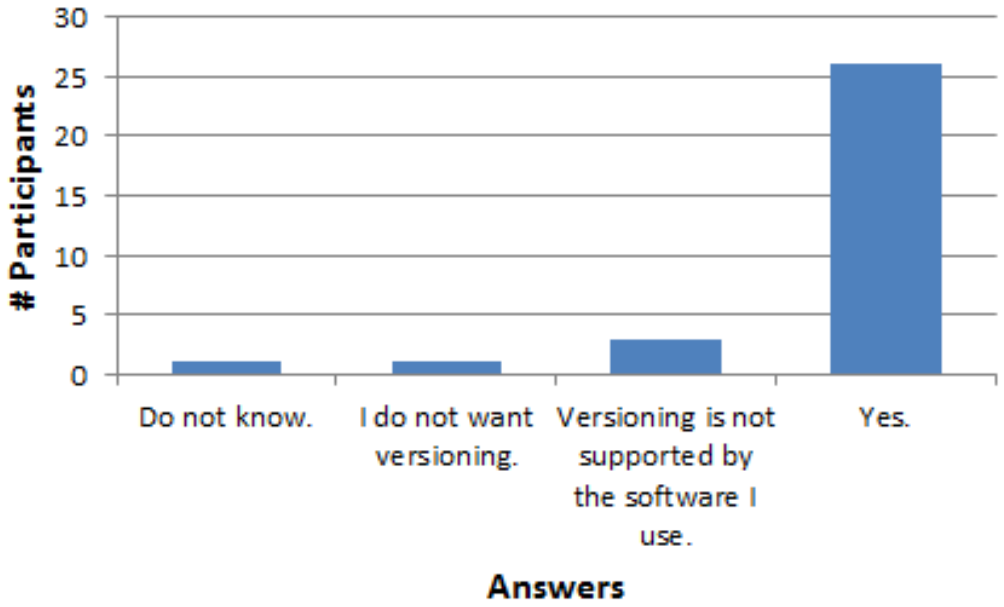


Figure 4.11: Do you use a versioning system?

contained inside their DMS, the results are shown in Figure 4.12. Only 26 of 31 participants use a DMS. Out of those 26 participants: one answered never, eleven answered 1%–25% of the time, five said 25%–50% of the time and nine participants said 50% of the time or more. To summarize, the majority of the participants work with documents that are not managed by their DMS at least 25% of the time.

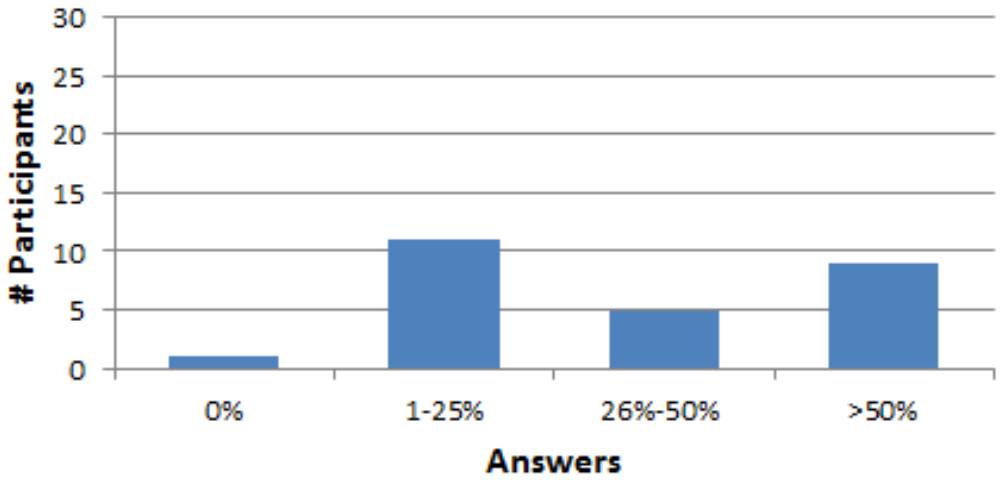


Figure 4.12: How often do you work with documents that are *not* stored inside your document management system?

Questions 7, 8 and 9

Of the 31 participants that answered the questionnaire, eight answered that their DMS supports the search of re-used content, as shown on Figure 4.13. The DMS used by those users were: GitHub via Golem, a wiki, Microsoft Sharepoint, Author IT and Google Drive. To our knowledge, those systems only support the general search of content, but not the specific search for content that has been re-used. We assume that the participants misunderstood the question. Of those participants with no support for searching re-used content, only two answered Question 8 (would you like to have support?) with no, whilst ten answered with yes. The other participants were undecided.

Question 9: "For which of your work processes are you utilizing the document management system?" was open-ended. We filtered the answers given by the participants into the categories from Question 1, with one exception, versioning. Versioning as a task includes information search, orga-

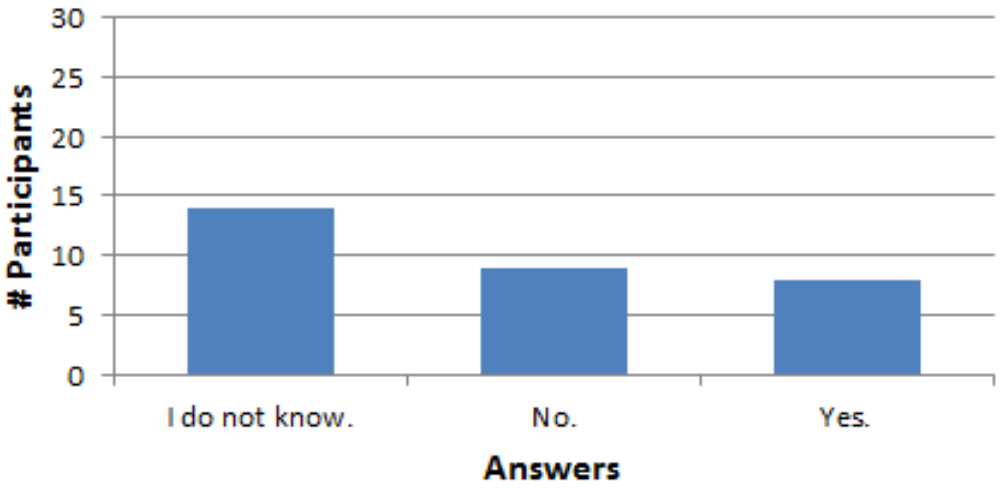


Figure 4.13: Does your content management system support search of re-used content?

nization, authoring and dissemination, but the participants chose to name it specifically. Figure 4.14 shows the results: 20 participants named versioning, nine information organization, seven information dissemination, seven (co)-authoring and six participants named information search.

Figure 4.14 also shows the difference between the number of times a task has been named in the Question 1 and the number of times it has been named in Question 9. We can see a drop of 50% or more for the tasks: Information Organization (9/19), (Co)-Authoring Information (7/24), Information Search (6/22) Disseminate information (7/21). The other tasks from Question 1 have not been named at all.

4.2.3 Analysis

This study had the goal of answering Questions A and B defined at the beginning of this section. We also wanted to gain further proof for the validity of the requirements we defined in Section 2.4.4.

Question A — Which tasks are the knowledge workers performing and are these supported by the DMS?

In Chapter 2 we introduced ten tasks that describe what knowledge workers do (Analyze Information, (Co-)Authoring Information, Acquisition of Information, Disseminate Information, Information Search, Information

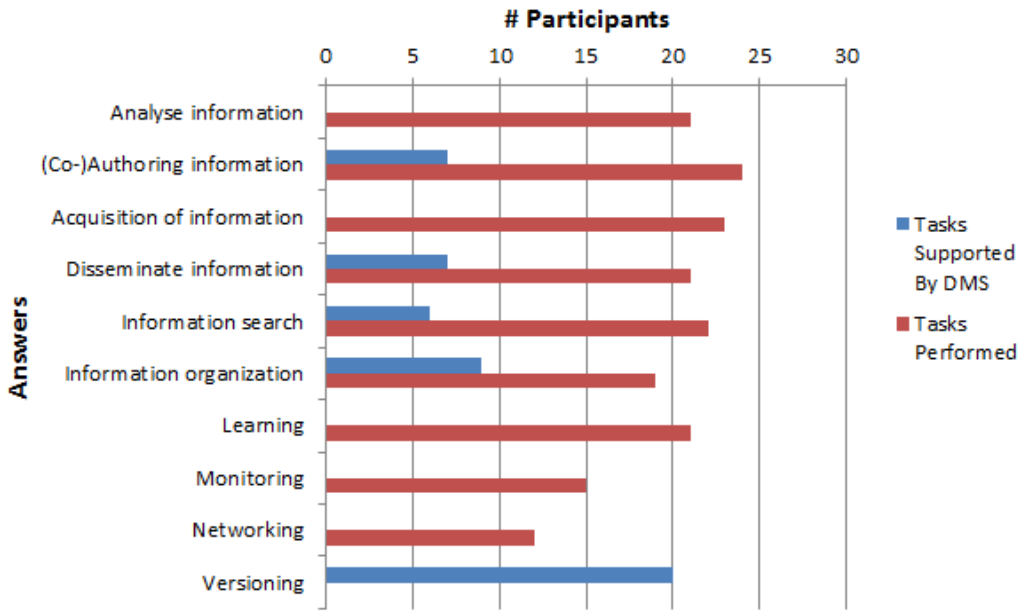


Figure 4.14: Number of Tasks Named in Questions 1 and 9.

Organization, Learning, Monitoring, Networking and Service search).

What Tasks? We identified four main tasks in Chapter 2 that knowledge workers need support with: (Co)-Authoring Information, Information Search, Information Organization and Information Dissemination. Figure 4.7 shows that these tasks were amongst the most named by the participants. Figure 4.14 further confirms that these four tasks are most important to consider, since these were the only tasks DMSs were used for. However, it is also clear that of all the participants naming these tasks, only a minority were able to utilize their DMS for support with these tasks.

Versioning Additionally, the majority of participants named the task *versioning* as being one that is supported by their DMS, as shown in Figure 4.11. This is interesting for two reasons, firstly that versioning is considered a task on its own even though it is a technical process that should be taken care of by the DMS. Secondly, versioning is an aggregation of the four tasks already named. This leads us to believe that the participants highly value the ability to version-control their data.

What issues? For the second part of the question:"... and are these supported by the DMS?", we found the answer to be no, the tasks are not fully supported. Although some of the participants named the tasks as being

supported by their DMS, they also often answered that they are working with data outside of the DMS, as seen in Figure 4.12. The second reason is the fact that DMSs do not support the search for re-used content snippets and only support versioning on file level, if at all. The first study has shown that content snippets are more important to users than files.

Question B — Do DMSs address the issues found in the first study?

The issues we found in the first study were: (a) maintenance of the file-folder system, (b) versioning of documents, (c) synchronizing documents, and (d) keeping track of instances of the same content.

Using a DMS seems to address issue (b) and (c), as 26 out 30 participants had access to versioning control, as shown in Figure 4.11. However, Figure 4.12 shows that all but one participant also work with documents outside of their DMS. This results in the support being incomplete. Issue (d) was not addressed by the DMSs named by the participants.

We therefore conclude that DMSs do not address the issues found in the first study.

Revisiting the Requirements

The results of the second study strongly support the first three requirements, for the following reasons:

Relationship Detection Versioning and Information Organization are two of the major tasks that participants seek support with. Both of these tasks are strongly supported by information about the relation two pieces of content have.

Relationship Metric Versioning and Information Organization again profit greatly from information about the relationship two pieces of content have, such as: ”which is the most recent version”. Additionally, Information Search and Dissemination profit from that knowledge, as users can access specific versions of content more easily if they can follow a trail of before-after connections. This is also of great help when (Co)-Authoring Information, as again, the latest version of content is often important, but hard to determine with the current tools available.

Distributed The study has shown that the vast majority of users works with content outside of their DMS. This fact supports our argument of a distributed solution, which enables users to access all their data when applying relationship metrics.

DMSs are often treated as the solution for knowledge workers' troubles regarding digital content organization and creation, but the previous study supports the conclusion that it is not. Additionally, we encountered potential study participants who wanted to introduce a DMS, but never managed to successfully do so. We decided to conduct a case study with these participants to learn more.

4.3 Case Study

We conducted a case study with a group of knowledge workers who unsuccessfully tried to introduce a DMS for their team. We aimed to shed light on issues accompanying the introduction of DMSs and to identify reasons why DMSs fail. The aim of this study is to discover the issues that arise when introducing DMSs. These issues are helpful when answering the third research question (How can content-centered provenance data tracking be implemented?), as we need to avoid these issues.

4.3.1 Study Design

We were invited by the participants' company to interview all six participants on site to ask their opinions on the different DMSs used. The interviews were guided by the time line of used DMSs, which was supplied to us beforehand. The interviews were not structured around particular questions, as the participants had different backgrounds and interaction levels with the DMSs used. However, we tried to answer two main questions for each DMS used:

1. How and why was it introduced?
2. Why did it fail?

We also asked each participant to rate the DMSs they were involved with on a Likert scale from 1 (bad) to 5 (good). The interviews were held in a meeting room and recorded, each interview lasted 30 minutes.

4.3.2 Results

We now present the results of the case study, starting with the statistics of the participants. We follow with a description of each DMS implemented by the work group of the participants. For each system we then answer Questions 1 and 2.

Participants

The target group consisted of six participants, aged 30–59 years old. We had three female and three male participants, identified with P1–P6. The roles of the participants were science (3), management (2) and librarian (1). However, we cannot allocate identifiers to the roles, since this would de-anonymize the participants due to the small sample size. The six participants were members of the same research work group at a local company. The document editing system implemented by the company was Lotus Notes. The company also hosts a shared network space called the Sky-Drive where the participants can store and share content.

Document Management Systems

The work group had used a wiki from 2007 to 2010 when a decision was made to introduce another DMS. From 2010 to 2014, four different DMSs were used with varying degrees of success. The DMSs were:

1. *Deki Wiki* was an open source web-based wiki solution developed by MindTouch until 2013.
2. *Open Atrium* is an architecture for Drupal that supplies a framework for a custom-built intranet to support collaboration and content management
3. *Alfresco* is an enterprise level content management system featuring a web portal, file system compatibility with Unix and Windows systems and social network components.
4. *Integrated Solution — Media Wiki, Wordpress Blogs, Forums*. This solution was built inside the work group with the aim to supply a custom single-login portal with access to all functionalities needed by the work group.

We asked all participants to rate each DMS on a Likert scale from 1 (bad)–5 (good), as seen in Table 4.1. One user chose to give two ratings for two systems. P2 chose go give Open Atrium a 1 (stability) and a 3.5 (usability) and the integrated solution a 2 (wiki component) and a 4 (WordPress component). We chose to use the lower rating, since P2 seemed to sway more towards the lower rating when discussing the system. Not all participants came in contact with all DMSs used (–).

	P1	P2	P3	P4	P5	P6	Average
DekiWiki	3	3	2	4	–	1	2.6
OpenAtrium	1	1	–	–	–	–	1
Alfresco	2	1	2	2	3	3	2.16
Integrated Sol.	2	2	–	2	4	–	2.5

Table 4.1: Likert Ratings for the Different DMSs Used, from 1 (bad) to 5 (good), – not applicable.

We now transcribe the process of implementing the different DMSs and the opinions on shortcomings and advantages of the different DMSs as perceived by the participants. We also give a recount of the participants' opinions on their current situation, which is not having a central DMS. They do have access to a shared network file-folder system: Skydrive.

Deki Wiki 2007–2010

How and why was it introduced? The Deki Wiki was introduced in 2007, the decision to chose that wiki was made by a member of the research work group not included in the study. All members of the research work group and the IT work group were given logins, including new employees. One participant (P1) stated that usage of this system was encouraged by company policy and by the fact that other people used it actively. One participant (P5) had not used the system at all. When asked, they were not aware it had existed. The Deki Wiki was deactivated in 2010.

Why did it fail? The system was liked by three, more tech-savvy, participants, as they praised its abilities (P2: "It was really good as it understood LaTeX", P2: "I used it quite a bit"). However, they noted that the system did have shortcomings, such as the lack of search capabilities for data inside the system. Those participants mainly used the system for informal and formal documentation of work processes and systems.

The other two participants disliked the system for several reasons, mainly because the interface was very basic and not intuitive to use (P3: ”I had to force myself to use it.”, P6: ”Re-finding old data was impossible.”), and that the effort needed to become efficient at using the system was to high. The decision to decommission the system was made at the company level, however the reasons were not clear for the participants. They speculated that the company wanted to introduce a company wide DMS. P4 noted that all data inside the Deki Wiki was lost when it was decommissioned.

Open Atrium 2010–2010

How and why was it introduced? Open Atrium was the first system considered to replace the Deki Wiki in 2010. It was suggested by an advisor of the research group as they had good experiences with it. It was tested for three months before the decision was made not to use it.

Why did it fail? P1 and P2 went into a three month trial period to set up and test the system. P1 and P2 said the main reason not to chose Open Atrium was the amount of software bugs found when trying to set up the system. The participants stated that a full time developer would have been needed to get the system running smoothly and the system was therefore not fit to be used.

Alfresco 2010–2012

How and why was it introduced? After the decision not to use Open Atrium, the research group introduced Alfresco. The system was chosen by P1 and P2 after creating a requirements list and researching the best fit for that list. After the initial set up of the system, every member of the group was given an account to use for the system. The system was decommissioned in 2012.

Why did it fail? The system got mixed feedback from all participants. The participants were praising the social functionalities (P1: ”It had Facebook like functions...”) and the more intuitive user interface. The participants noted several shortcomings when asked about the system.

The first shortcoming was a technical error that sometimes resulted in the permanent closing of an account, resulting in the loss of the ability to edit the data attached to that account. This was a major fault within the

system (P2: ”You do that once or twice, but then you are done.”, P6: ”I stopped using the system after I got locked out.”).

The second shortcoming was that not all members of the work group embraced and used the system (P4: ”I certainly used it less than the Deki Wiki”), resulting in frustration for the participants who used the system, as they would have to double their efforts to reach members of the work group not using the system. Additionally, P6 noted that every participant was left to create their own structure for holding their data, resulting in ’wild growth’ style structures that were hard to use for other participants. The system was decommissioned when most members of the work group stopped using it.

Integrated Solution 2012–2014

How and why was it introduced? P2 decided to provide an independent solution (maintained only by P2, outside company reach) after Alfresco was decommissioned. They introduced an integrated solution that combined a MediaWiki, WordPress blogs and a forum into one system. P1 stated that after the initial setup, every member of the work group was given an account and a challenge to complete basic tasks to get the users started in the new system. This system was decommissioned at the beginning of 2014.

Why did it fail? P1 stated that most users never used the system for more than the initial challenge. The participants did not see any advantages of this solution over Alfresco functionality wise, but noted that some of the main disadvantages were the same, most importantly the lack of users (P5: ”There still was no feedback on whether my content was used by other people or not.”). Only four out of six participants used the system. The system was decommissioned because it was not being used anymore by work group members.

2014 and Ongoing — SkyDrive

All participants stated that the current situation is not ideal (P1: ”There are a lot of people in the company that if they got hit by a bus tomorrow, there would be a great big black hole.”, P2: ”We should have invested in a proper DMS 10 years ago.”). The SkyDrive is used by Participants 3, 5 and 6. However, these participants also stated that SkyDrive does come

with its own issues, mainly the lack of collaboration support (collision of editing times for content) and the fact that the structure is user-maintained, which leads to all the issues we found with the file-folder system. However, P3 stated that this was still a better solution, since they could be sure the system would not vanish one day and the data would be lost (as happened with the previous systems).

4.3.3 Analysis

We aimed to answer two questions for each DMS with this case study: (1) How and why was it introduced? and (2) Why did it fail?. We wanted to answer these questions as the results would be useful insight to answer the third research question (How can content-centered provenance data tracking be implemented?).

How and why was it introduced?

The decision to introduce each DMS after the first (Deki Wiki) was a result of the previous system failing. The selection process of the DMSs was mainly driven by two participants (P1 and P2), with varying degrees of input from the research group. For example, a requirements list was created for the third system, whilst the last system was built according to P2's past experiences. We found that the introduction process of a DMS can have an influence on its success. This is shown by some participants (P3 and P4) voicing frustration over not knowing which system will be used for how long. However, these issues are of a social nature and therefore not targeted by our approach.

Why did it fail?

We found that the introduced systems failed for the following reasons:

1. The targeted users had different technical backgrounds and skills, resulting in different expectations for the usability of the system.
2. All systems but the first lacked user saturation, which lead to the participants using the system being frustrated. This issue was partly a result of a system switch in the first place, as users were hesitant to put effort into a system that might be decommissioned again.

3. Although the users worked in the same work group, their daily tasks and work processes differed significantly. They therefore had different requirements on what tasks a system should support and what features it should have.

To successfully answer Research Question 3, these issues need to be avoided when designing a DMS.

Revisiting the Requirements

The first two points are addressed by Requirements 3 and 4. A distributed automated system achieves full user saturation by default and frees the user of the effort to 'actively' feed it the information it needs to be useful. Such a system is therefore better suited for users of all technical backgrounds. We can therefore conclude that our requirements are suitable for designing a system avoiding the issues found when introducing a DMS successfully.

4.4 Summary

We conducted three user studies, which confirmed the issues previously found with the file-folder system. We used the answers gained to verify the four requirements defined in Section 2.4.4. We know describe how the results of our studies contribute to answering Research Questions 1 and 2 (Study 1 and 2) and Research Question 3 (Study 3).

4.4.1 What tasks do knowledge workers perform when working with digital content?

Study 1 The first study was aimed at discovering general issues knowledge workers have when working with digital documents. The tasks participants had issues with were: versioning, synchronization and keeping track of re-used content.

Study 2 The second study confirmed that the most important tasks to support are: (Co)-Authoring Information, Information Search, Information Organization, Information Dissemination and Versioning. The initial list of tasks did not include versioning as one of them. However, we decided to include the task after both Study 1 and 2 had participants specifically

mentioning versioning.

4.4.2 What are the issues with the current used systems aimed to support knowledge work?

Study 1 The first study confirmed ongoing issues with the use of the file-folder system, mainly:

a. High maintenance.

b. Keeping track of versions.

c. Synchronizing files kept in different physical locations.

d. Keeping track of instances of the same content across files.

Issues (a), (b) and (c) are targeted by the Requirements 3 and 4, as an automated and distributed system relieves the user of the maintenance work and keeps track of the versions, independently of where they are stored. The last issue is a clear indication towards the need to track provenance data on the content level instead of the file level, confirming the Requirements 1 and 2.

Study 2 Issues (a), (b) and (c) found in the first study are targeted by enterprise DMSs. The second study therefore targeted knowledge workers with access to DMSs, to see if the issues connected with the file-folder system are resolved. We also aimed to learn which tasks knowledge workers seek support with when using a DMS. The results of the second study confirmed that DMSs are not solving the issues found in the first study.

Issues (b) and (c) are not solved, since all participants still worked with content stored outside of their DMS, therefore rendering the gained metadata and organizational efforts by the DMS incomplete. The amount of maintenance (a) is also effected by this, since users need to organize files inside and outside of the DMS. No DMS named was able to keep track of re-used instances of the same content inside the DMS, much less content outside of it. This confirms Requirements 3 and 4, as a successful system needs to be distributed and automated to avoid the shortcomings detected in this study.

4.4.3 How can content-centered provenance data tracking be implemented?

Study 1 and 2 The first two studies confirmed our choice of supported tasks. We also discovered what issues our content-centered provenance data tracking system needs to address. The last study was conducted to discover issues related to the nature of DMSs and their introduction to the work place, instead of issues related to the tasks users performed.

Study 3 Whilst conducting the second study, we encountered participants who tried implementing DMSs into their work environment with no success. We decided to follow up in order to learn about the issues involved, as these might be valuable pointers towards a better system to help the knowledge workers. The last study was a case study conducted at the research group of a local company. The research had tried to implement different DMSs with varying degrees of success. We found three main reasons for the used systems not being successful:

A Low user saturation: The system was only used by a small portion of the users who had access to it.

B Differing levels of user experience / tech affinity lead to differing expectations towards the usability of the DMS.

C Differing expectations in regards to supported features.

Issues A and B are directly targeted by the Requirements 3 and 4, since an automated distributed system has maximum user saturation and a minimum of required user knowledge to be used. Requirements 1 and 2 target issue C, since all involved participants were involved handling content one way or another and would therefore be aided by the gained information. We have now fully answered the first two research questions and partially answered the third research question:

1. What tasks do knowledge workers perform when working with digital content?
2. What are the issues with the current used systems aimed to support knowledge work?
3. How can content-centered provenance data tracking be implemented?

In the next chapter, we aim to fully answer the third question: How can content-centered provenance data tracking be implemented?

5

Document DNA Model

After establishing a list of requirements in Section 2.4.4 and confirming these requirements through studies reported in Chapter 4, we now introduce the Document DNA (DDNA) model. This is a provenance data annotation model, answering the third research question.

We start by briefly sketching the concept of DDNA and reviewing biological DNA and comparing it to the requirements. We then introduce DDNA in detail by specifying the concepts of documents, document states, actions and sessions. Actions are used to define the minimum level of detail we track when working with content, while sessions are used to define the interval at which provenance needs to be updated.

We define relations between two documents and possible queries based on the defined relations. We also give some scenarios and examples for the application of the defined queries. Finally, we discuss how this design fulfills the requirements set in the earlier chapters: Relationship Detection, Relationship Metric, Distributed and Automated.

5.1 DNA and DDNA

We here introduce the DDNA concept and compare its characteristics with those of the DNA found in life-forms.

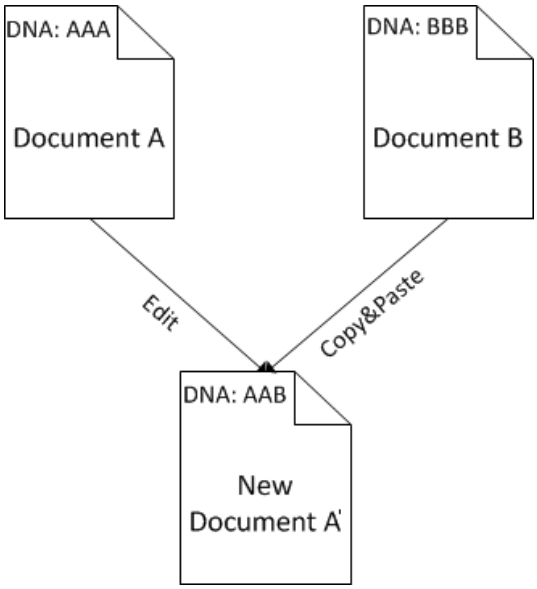


Figure 5.1: Example of Documents and their Document DNA

5.1.1 DDNA Concept Sketch

Our provenance data annotation model supports the tracking of content snippets by attaching a signature directly to the content of a document. When the content is manipulated, the signature is changed to reflect the changes made to the content. A document holds together with its content all signatures off previous changes.

Figure 5.1 illustrates an example of DDNA: Content in Document A is edited, and also some content from Document B is copied and pasted into Document A. Therefore, the DDNA of the new Document A' is a mixture of the DDNAs of the documents A and B that contributed content to A'. In this example, two sources of content are used to create a new instance of content, which is similar to how the way new life-forms are created by combining the DNA of both of their parents. This characteristic gave rise to naming of the DDNA model.

5.1.2 Comparison of DNA and Requirements

In this section, we first compare the conceptual characteristics between DNA and DDNA Requirements. We then examine the current algorithms used to compare different DNA strings to see if those algorithms are suitable for our approach. For this comparison, we regard life-forms as docu-

ments, with DNA being the equivalent to document annotations.

Relationship Detection The first requirement is fulfilled, DNA matching can be used to determine if two life-forms are related. For example, one could detect if two people are brother and sister.

Relationship Metric This requirement is also fulfilled. Different traits of life-forms are represented by their DNA. Therefore researchers can compare two DNA strings and identify what traits are shared. The methods used for this will be discussed in Section 5.1.2. One example is inheritable diseases, such as those that can be predicted by analyzing the DNA of a human.

Distributed & Automated Since DNA is part of every life-form and created newly whenever a new life-form is born, the last two requirements are also met.

Since our approach targets documents, one could argue that information in documents can be influenced by many different documents, whereas the DNA of life-forms is influenced by one (cell division or cloning) or two DNA strings (reproduction of complex life-forms) of their direct predecessor. However, recent research has shown that the DNA of bacteria can be influenced by many other bacteria, through horizontal gene transfer (de la Cruz and Davies, 2000). This includes bacteria many generations apart. We conclude that there are enough similarities between DNA and DDNA to warrant exploring methods of DNA string comparison.

Detecting Relations Through DNA — Algorithms

Here we discuss current methods of biologists for the comparison of DNA of different life-forms since they might be useful for our approach. Biologists (Lesk, 2013) represent the nature of relationships between different life-forms by using phylogenetic trees (directed graphs). In these trees, the direct ancestor of a life-form has an outgoing edge to the descendant life-form, as seen in Figure 5.2. These edges represent that (part of) the DNA has been transferred from ancestor to descendant. To create those graphs, either clustering or cladistic methods are used.

Lesk states that cladistic methods are superior to clustering methods for creating those trees. Therefore, we limit our discussion to those methods.

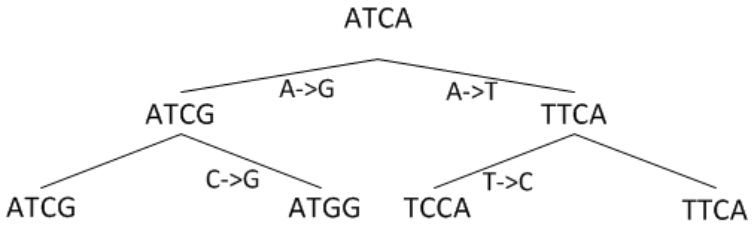


Figure 5.2: Example of a Phylogenetic Tree as in Lesk (2013)

Figure 5.2 illustrates a phylogenetic tree for four species represented by the DNAs: ATCG, ATGG, TCCA and TTCA. The edges in the figure represent mutations that need to happen between ancestor and descendant. Note that this is only one possibility for such a tree for those life-forms.

Maximum Parsimony The maximum parsimony method generates the phylogenetic tree so that it includes the minimum number of mutations, like the tree in Figure 5.2.

Maximum Likelihood The maximum likelihood method weights the possibility of mutations and creates the tree so it shows the path of the most likely mutations.

At first, these methods seem suitable for our approach. However, we detected several shortcomings, with regards to this research, that we need to address.

Mutation Rates Lesk states that varying rates of evolution bring additional issues when creating the phylogenetic trees. In such a case the mutation rates need to be known before creating the tree. The pace at which digital content changes can differ greatly between versions, which means that this would cause issues when calculating phylogenetic trees.

Probability vs Observation The introduced methods rely on probability for the creation of a phylogenetic tree. For example, it might be more probable that the string ATCG first mutates to ATGG and then AAGG. Such probabilities do not exist for digital content. In addition, we are able to observe the content when it is created and manipulated, removing the need for such probability.

Horizontal Transfer Horizontal gene transfer is not supported when creating phylogenetic trees. However, horizontal content transfer between two pieces of digital content must be supported. For example, a user could edit

a figure several times and through different versions of a piece of content, but then decide to go back to the original version of the figure.

In summary, current methods used for comparing strings of DNA are not suitable for our approach, for the following reasons:

- They rely on statistics of evolution and mutation, which do not exist for digital content.
- They aim to decode the DNA after it was created using probability, whereas the creation and manipulation of digital content can be directly observed.
- They do not support horizontal gene transfer, but it will occur when editing content and therefore needs to be supported.

We conclude that the use and characteristics of DNA make it a suitable metaphor for describing our approach and that phylogenetic trees are suitable as a way of displaying relationships. However, the methods used to create those trees cannot be directly applied to digital content. The DNA therefore provides an inspiration in this work, rather than a blueprint.

5.2 DDNA Model

We introduced a rough sketch of our approach at the beginning of the chapter. This section now introduces the DDNA model through step-wise definitions of its components. We also provide examples and illustrations for each of these definitions.

5.2.1 Document

To address Requirement 3 (Distributed), metadata about content needs to be stored with the content. Documents are containers holding digital content. Since content cannot exist outside a container, it is sufficient to store the metadata in the container that holds the content. Our model defines digital documents as a triple of an object O, the content C and a temporary history Z.

Definition 1 (DDNA Document)

A DDNA Document D represents a digital document and is defined as a triple $D = [O, C, Z]$ *of object O, content C, and history Z.*

- *The object O is the container holding any (system specific) information concerning the content, including format.*
- *Content C is the information contained in the document, stripped of formatting or style. C represents a set of content pieces* $\{C_1, ..., C_m\}$*, with cardinality* $|C| = m$*.*
- *The temporary history Z is a sequence of tuples* $Z = ([A_1, R_1], ..., [A_n, R_n])$ *with Actions* A_i *(as per Definition 2) and References to the position and length of the manipulated content* R_i $(1 \leq i \leq n)$*, with cardinality* $|Z| = n$*.*

The set of all documents is denoted $\mathbb{D}$*. An empty document is identified by empty content C.*

Document Example

We consider a text document (.txt) containing the sentence: "This is an example." This text document is represented within the DDNA model as Document D with the following components:

- O is the .txt document container, including properties such as document size (4 kB), character count (19), and position of each C_i within the document (position of C_1:0).
- $C = \{C_1\}$ is the text block "This is an example."
- Z is the history of actions that led to the creation of the text block "This is an example." After typing this sentence, the last entry in Z is $[\text{insert }., R_{19}]$.

Throughout the remainder of this chapter, the term document refers to DDNA Document, unless stated otherwise.

5.2.2 Action

Actions are basic activities that users apply to digital content.

Definition 2 (Actions) *An Action is a function* $A : \mathbb{D} \to \mathbb{D}$*, with* $A(D) = D'$*. The following actions are defined in the model:*

- *Insert content* — A_i
  $A_i(D) = A_i([O, C, Z]) = D' = [O', C', Z']$
  with $O' = O+$ *details and format of inserted content,* $C' = C+$ *inserted content, and* $Z' = (Z_1, ..., Z_n, [A_i, R_{n+1}])$.

- *Delete content* — A_d
  $A_d(D) = A_d([O, C, Z]) = D' = [O', C', Z']$
  with $O' = O-$ *details and format of deleted content,* $C' = C-$ *deleted content, and* $Z' = (Z_1, ..., Z_n, [A_d, R_{n+1}])$.

- *Format content* — A_f
  $A_f(D) = A_f([O, C, Z]) = D' = [O', C', Z']$
  with $O' = O+$ *details and format of manipulation,* $C' = C$, *and* $Z' = (Z_1, ..., Z_n, [A_f, R_{n+1}])$.

- *Select content* — A_{se}
  $A_{se}(D) = A_{se}([O, C, Z]) = D' = [O', C', Z']$
  with $O' = O, C' = C, Z' = (Z_1, ..., Z_n, [A_{se}, R_{n+1}])$.

- *Copy content* — A_c
  $A_c(D) = A_c([O, C, Z]) = D' = [O', C', Z']$
  with $O' = O, C' = C$, *and* $Z' = (Z_1, ..., Z_n, [A_c, R_{n+1}])$.

- *Paste content* — A_p
  $A_p(D) = A_p([O, C, Z]) = D' = [O', C', Z']$
  with $O' = O+$ *details (DDNA) and format of pasted content,*
  $C' = C + C_{m+1}$, *and* $Z' = (Z_1, ..., Z_n, [A_p, R_{n+1}])$.

- *Save document* — A_{sa}
  $A_{sa}(D) = A_{sa}([O, C, Z]) = D' = [O', C', Z']$
  with $O' = O, C' = C$, *append* Z *to the DDNA signature of the document, and* $Z' = \emptyset$

- *Cut content* — $A_c + A_d$

The DDNA itself is defined in Section 5.2.5.

Insert & Manipulate The insert and manipulate actions are always connected to one piece of content C_m contained in the document D (the content the action was performed on).

Select, Delete & Copy The select, delete and copy actions may be connected to one or more pieces of content, since they are performed on selections that might include more than one piece of content. Select may also be applied to empty content, which represents un-select.

Paste The paste action is the only action that can add a new piece of content C_{m+1} to the document D. It therefore adds a new DDNA instance to O.

Open & Close Opening and closing a document are not regarded as actions, since nothing is to be learned from these actions.

DDNA(s) is created or changed when a document is saved, using the information collected in Z. The DDNA is content-centered, meaning that a Document D will contain a DDNA in O for each piece of content C_m included in C. The save action $A_{sa}(D_x)$ always results in a new document D_y. All other actions result in a document state between saved documents, as explained in Section 5.2.3. We do not consider format and format changes to be part of content evolution. Therefore, format changes are not defined as an manipulation action. Undo is not considered a separate action, as the document simply reverts to the last state before the undone action was taken. The undone action is removed from the history and the DDNA is not changed.

Action Example

Actions on their own are self-explanatory. However, actions used in sequence are covered in Section 5.2.4.

Actions and ACID

Since actions are comparable to the concept of transactions in databases, we compare the properties of actions and transactions. We do so by checking for the ACID requirements on actions.

Atomic All actions are atomic, meaning that actions cannot be half completed, since this could result in a corrupt document. For example, an update in the content but not the object would lead to an incorrect character count.

Consistency Actions can lead to an inconsistent state. This means that the consistency requirement is not met.

Isolation All actions are executed strictly sequential (one after the other), therefore no action can interfere with another action. This means the isolation requirement is met.

Durable A committed action will only be durable if the document is saved afterward and if the action is neither select or copy. All actions are durable as long as the document is opened and the work process is ongoing.

We can conclude that actions do not follow the ACID principle, since consistency and durability are not guaranteed.

5.2.3 Document States

Document states describe the state a document is in according to the last action executed on it. Figure 5.3 provides a graphical depiction of the states: temporary, consistent, saved, and acts as a legend for other figures.

Figure 5.3: Document States

Definition 3 (Document States) *Three document states are distinguished depending on the last (n^{th}) entry in Z.*

- ***Temporary State**: $Z_n = [A_{se}, R_n]$ or $Z_n = [A_c, R_n]$. — The document is in the temporary state if the last entry in Z refers to either a selection or copy action, as these are not stored when the document is closed.*
- ***Consistent State**: $Z_n \neq [A_{se}, R_n]$ and $Z_n \neq [A_c, R_n]$. — The document is in a consistent state if the last entry in Z refers to an insert, delete, format, or paste action.*
- ***Saved State**: $Z = \emptyset$. — A document is in a saved state if the history is empty, i.e., after saving the document.*

A freshly opened document is in the saved state. Selections or content held in a copy buffer will not be restored when reopened. All other actions transform a document into the consistent state. The consistent state represents what the document will look like when it is saved, closed and reopened later on.

Document States Example A user opens Document T, which is a letter template. T is in the saved state when opened. The user now adds a specific address to the template, which transfers T into the consistent state. The address is then selected by the user and copied to be used in another document. T is now in a temporary state since the last action in Z is a copy. Finally, the user saves and closes the letter, which puts T in the saved state again.

Actions and States

When actions are applied to content, the state of the document holding the content may change or not change, depending on the action. Figure 5.4 illustrates the state transitions of two documents for several applied actions, using a finite state machine model. When content in one document is copied, the copied content is held in a separate content buffer, like the Microsoft Clipboard. This content can be pasted into another document by applying the paste command on the document. The buffer also contains the DDNA connected to the content.

5.2.4 Session

Sessions are used to describe a completed set of actions.

Definition 4 (Session)

A document D can only be assigned to one Session S at any time. A session S is a tuple $S = [D, AL]$ *with* $D \in \mathbb{D}$ *and a sequence of actions* $AL = (A_1, ..., A_n)$ *executed on D. If an action* $A \neq A_{sa}$ *is executed on D and no session is currently assigned to D a session is created. If a save action* A_{sa} *is added to AL, the session is closed and a closed session cannot be re-opened. If at any given time two or more Sessions* $S_1, ..., S_n$ *exist, all executed actions are added to all lists* $AL_1, ..., AL_n$.

A session has one or more starting documents. D is the initial starting document of a Session S. Any other document D^* *is also considered a starting doc-*

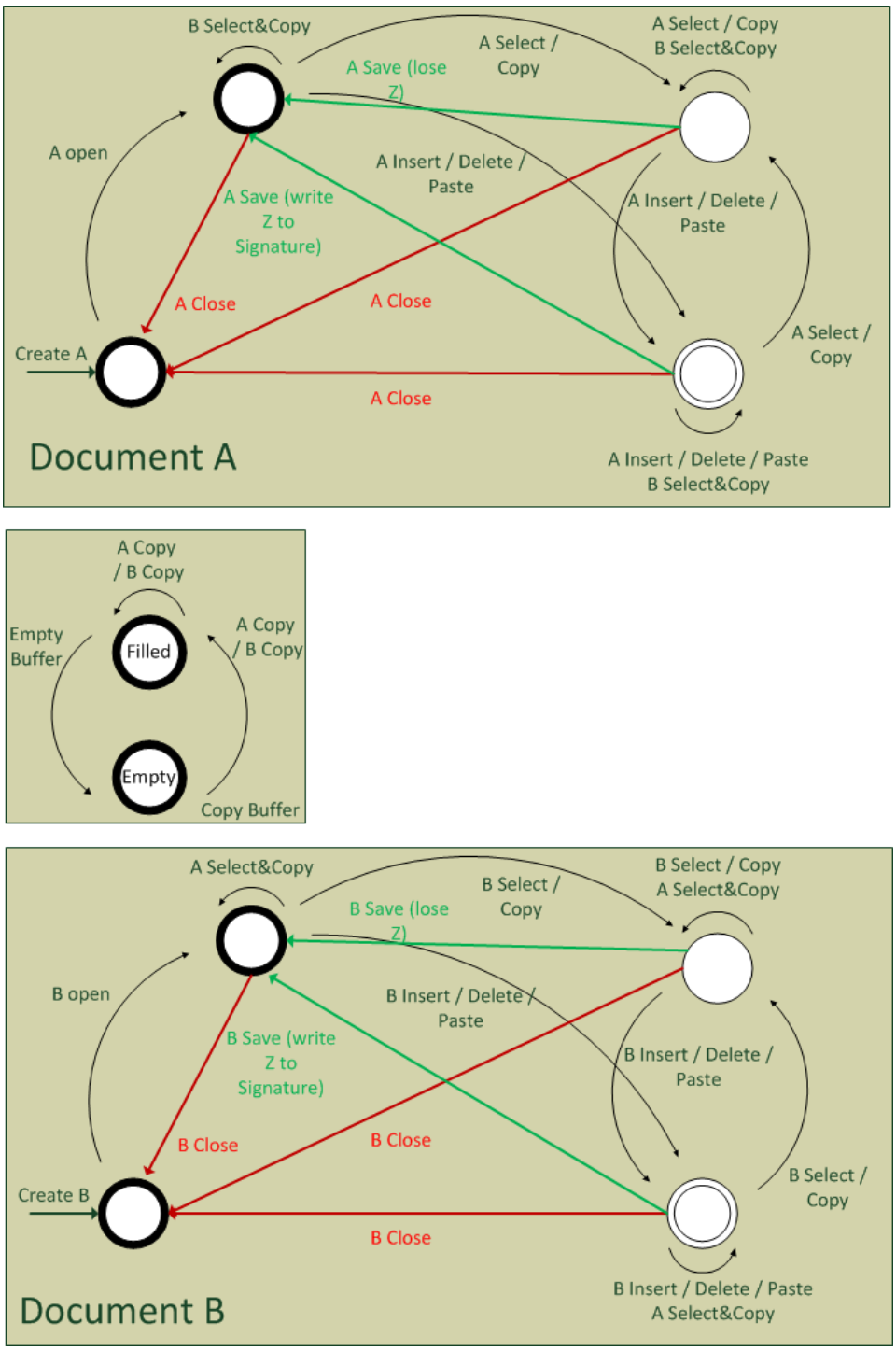


Figure 5.4: Actions and States

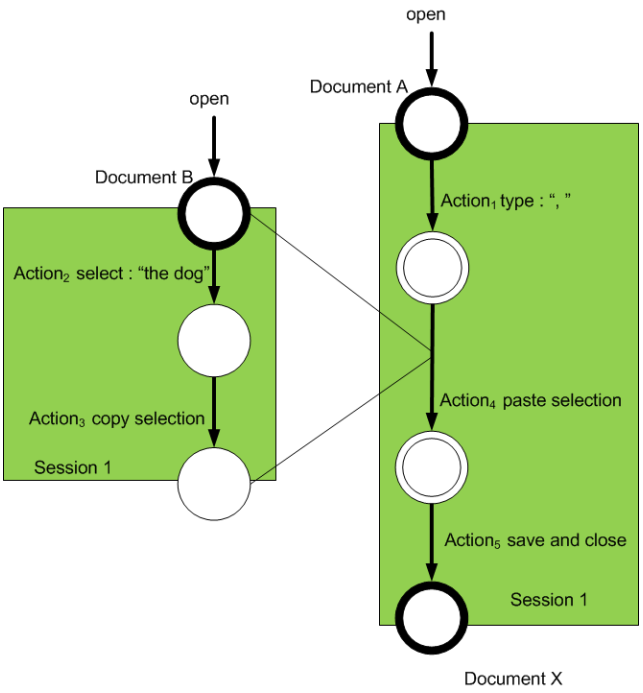


Figure 5.5: Session 1

ument of S if AL includes a pair of copy and paste actions $A_c(D^*)$ and $A_p(D)$, where AL does not contain any copy actions between these two actions. All starting documents for a session S are defined as the set SD. A session has exactly one ending document ED, which is the new document D' that is created by the save action $A_{sa}(D) = D'$.

A session starts and ends with a saved document. There are at least two documents involved in one session, a starting document and a saved document at the end. However, the number of starting documents is not limited. Every document contributing a copy and paste pair of actions in a session, that is not the saved document at the end, is a starting document. We now introduce some examples to clarify the concept.

Session and States Examples

Figure 5.5 illustrates a session which is starting with Documents A and B and results in Document X. X is the result of a minor change to A's content and the insertion of some content from B. Figure 5.6 illustrates two overlapping Sessions. Session 1 is the same session as in Figure 5.5. Session 2 was finished by changing the content of Document B and saving after the copy action. Note that Actions 2 and 3 are part of Session 2, since Actions

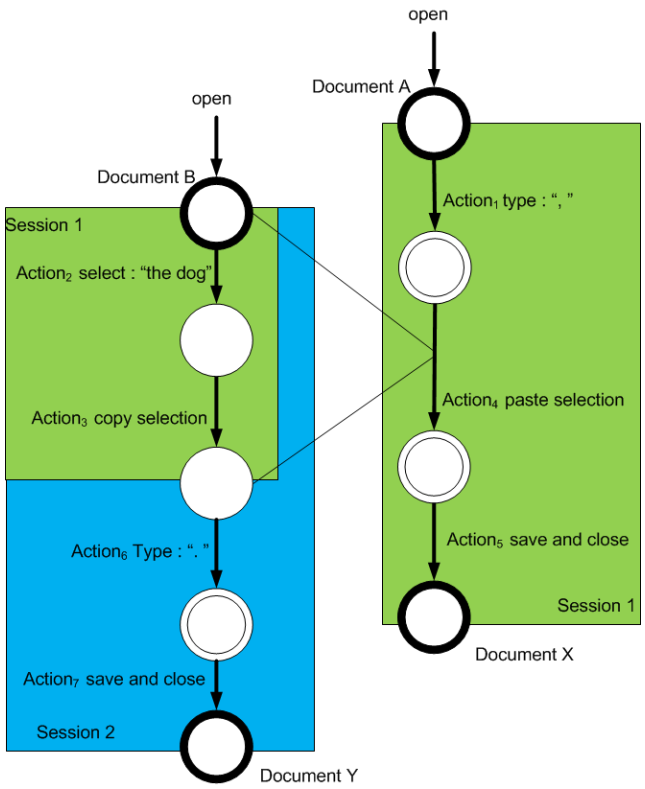


Figure 5.6: Sessions 1 and 2

2 and 4 are a copy and paste pair with the paste action executed on the starting document of Session 2.

This save action results in the saved document's DDNA to be adapted, creating a new set of DDNA. This adaptation of the DDNA(s) represents the relation the content included in the saved document has to the content included in the starting documents. The new set of DDNA is created by processing the actions that are recorded in Z.

Session and Documents Examples

Figure 5.7 illustrates the impact of the actions on the documents for Session 1. In this figure, the actions executed for Document B do not lead to a new document, since no content is manipulated and no save action is executed. This means that when Document B is closed, Actions 2 and 3 only remain in Document X's history and therefore Document X's DDNA.

Figure 5.8 illustrates Sessions 1 and 2 and the actions involved. In this example, Actions 2 and 3 are processed two times, firstly when Document X is created and secondly when Document Y is created. The two actions do

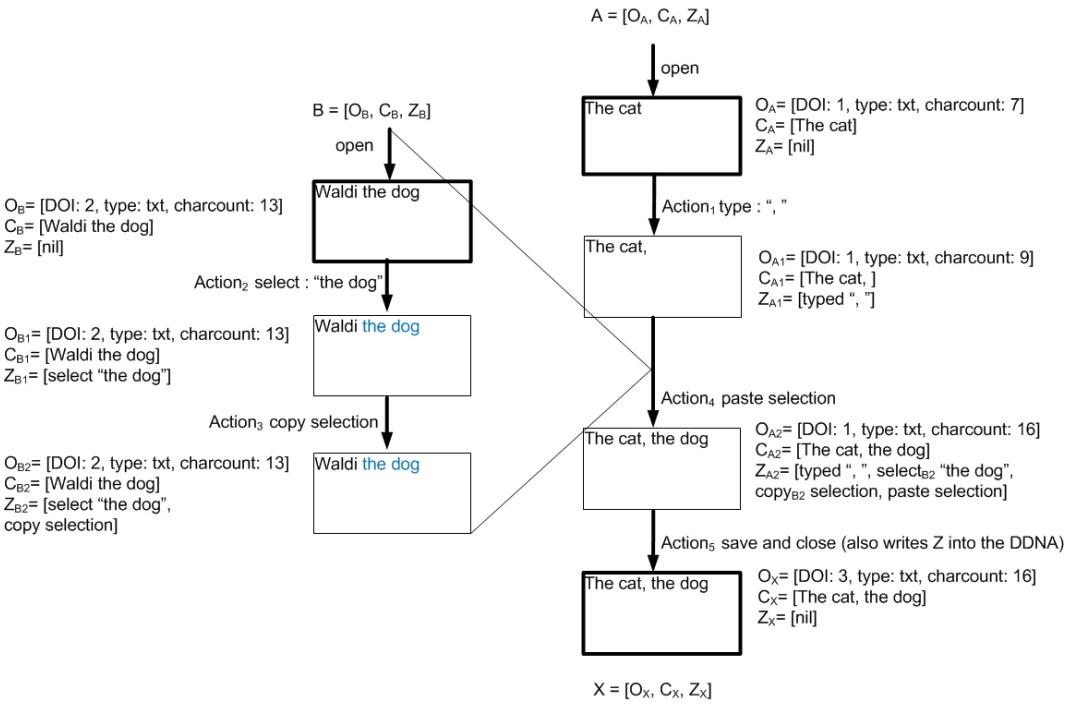


Figure 5.7: Actions for Session 1

not have any impact on Document Y's DDNA, since they did not trigger any content change.

5.2.5 DDNA Signature

We now define the DDNA Signature.

Definition 5 DDNA Signature
A DDNA Signature $DDNA_{C_i}$ for a content piece (included in a Document D, $C_i \in C_D$) is a tuple $DDNA_{C_i} = [I_{C_i}, L_{C_i}]$, with I_{C_i} being a unique Identifier, $i \neq j \Rightarrow I_{C_i} \neq I_{C_j}$ and L_{C_i} being a Sequence of action lists $L_{C_i} = (AL_k, ..., AL_l)$. Once a session $S = [D, AL_S]$ with the ending document $ED = [O_{ED}, C_{ED}, Z_{ED}]$ is closed, a new action list AL_n is added to L_{C_i} if $C_i \in C_{ED}$ and there is a insert, delete or paste action referring to C_i in AL_S. AL_n includes all insert, delete and paste actions $A \in AL_S$ that refer to C_i.

I allows to uniquely identify each DDNA assigned to content and preserves the relation between copied and pasted content. The sequence of action lists L reflects the evolution of the represented content.

DDNA Signature Example

Following our example introduced in Subsection 5.2.1, we consider a saved text document (.txt) containing the sentence: "This is an example." where $C = \{C_1\}$ is the text block "This is an example." The DDNA Signature DDNA_{C_1} consists of the Identifier $I_{C_1} = 1$ and the Sequence of action lists $L_{C_1} = (AL_1)$, where the last entry in AL_1 is (insert .).

5.2.6 Content Relations

Content can be related in two ways: the ancestor/descendant-relation and sibling relation.

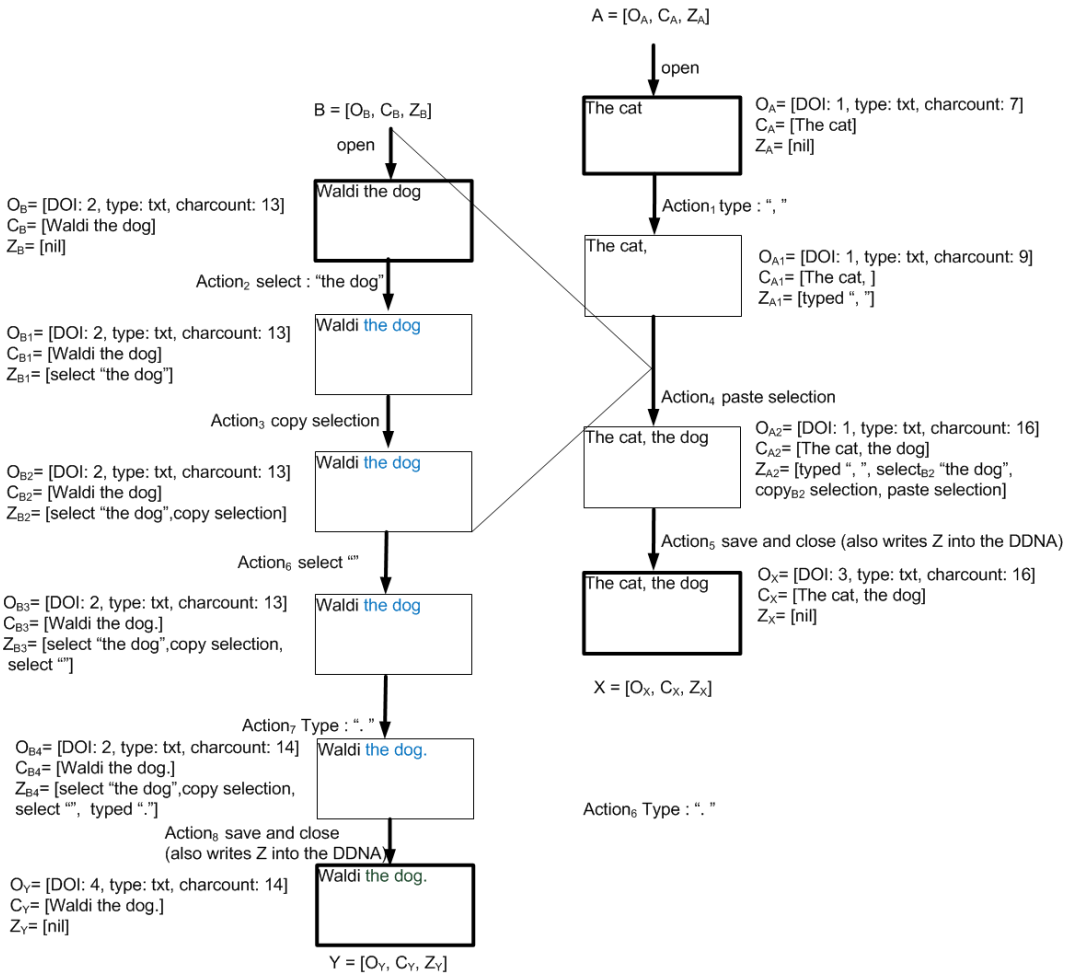


Figure 5.8: Actions for Sessions 1 and 2

Ancestors and Descendants

Ancestors and descendants relations are used to describe earlier or later versions of the same piece of content.

Definition 6 (Ancestor/Descendant Relation) *Let* $A = [O^A, C^A, Z^A]$ *and* $B = [O^B, C^B, Z^B]$ *be two documents.* $C_i^A \in C^A$*, with the identifier* I_{A_i}*, is said to be ancestor of* $C_j^B \in C^B$*, with the identifier* I_{B_j}*,* $(C_i^A \neq C_j^B)$ *and* C_j^B *is said to be descendant of* C_i^A*, if there exists a session S with* $A \in SD$ *and* $B = ED$ *and* $I_{A_i} = I_{B_j}$*. This relation is transitive but not symmetric.*

Ancestor Relation In order for Content A to be an ancestor of Content X, Content X itself or one of its ancestors must have been a direct result of a series of actions performed on Content A. This means that a piece of content may have an unlimited number of ancestors.

Descendant Relation In order for Content X to be the descendant of Content A, Content X or one of its ancestors must have been a direct result of a series of actions performed on Content A. A piece of content may have unlimited descendants.

Youngest & Oldest Property To be the youngest descendant of Content A, Content X is not allowed to have any descendants itself. To be the oldest ancestor of Content X, Content A is not allowed to have any ancestors itself. Note that it is possible that several pieces of content can be the youngest descendants, but only one piece of content can be the oldest ancestor.

Ancestor/Descendant Relation Example Figure 5.5 illustrates a session that leads to a new document. Documents A and B contain content that is the ancestor of content contained in Document X, since the content from both Documents A and B is manipulated through a series of actions and then stored in Document X.

Siblings

The sibling relation is used to describe two pieces of content that share an older version of themselves.

Definition 7 (Sibling Relation) *Let* $A = [O^A, C^A, Z^A]$*,* $B = [O^B, C^B, Z^B]$*, and* $C = [O^C, C^C, Z^C]$ *be three documents.* $C_i^A \in C^A$*, is said to be sibling of* $C_j^B \in C^B$

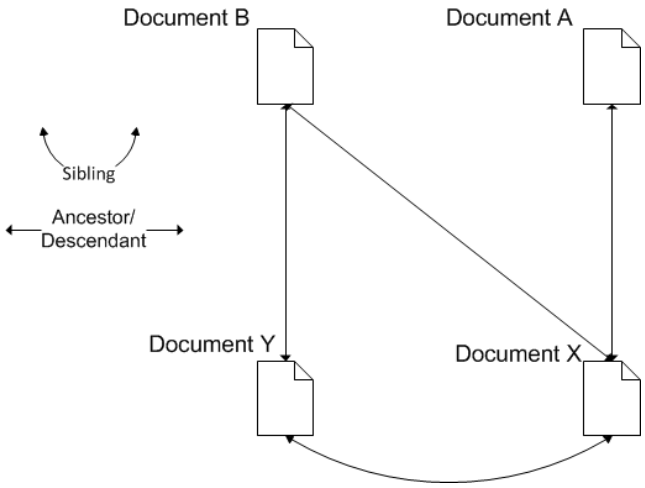


Figure 5.9: Relations between Documents

if there exists a C_k^C *with* $(C_i^A \neq C_j^B \neq C_k^C)$ *and* C_k^C *is ancestor of both* C_i^A *and* C_j^B*. The sibling relation is transitive and symmetric.*

Sibling Relation Example Session 2, illustrated in Figure 5.6, leads to Documents Y and X including contents that are siblings. Figure 5.9 illustrates all relations between the contents included the Documents A, B, X and Y that result from the Sessions 1 and 2.

Content Relations and Documents Relations

Since content is always held in a document, we need to look at how the relations regarding content translate to documents. Documents including pieces of content that are siblings or ancestors/descendants of other content inherit these relations. However these inherited relations between documents are not transitive, as the following two examples illustrate:

Document Relations Example Figure 5.10 illustrates an example including Documents A, B, C, D, E and F where the sibling relation between documents is not transitive. Documents A and B both include a single piece of content, C_{a1} and C_{b1}. Document D is created with its own piece of content C_{d1} and also copied content C_{a2} from A. Document F is created with its own piece of content C_{f1} and some copied content C_{d2} from D. Although A is ancestor of D, and D is ancestor of F, A is not ancestor of F, because A and F include no pair of content pieces with that relation. Document C is created with copied content C_{a2} from A and C_{b3} from B and some original content C_{c1}. Document E is created with original content C_{e1} and copied

content C_{b2} from B. Although Document C is a sibling to both D and E, D and E are not siblings to each other. The reason for this is that D and E include no pair of content pieces that are siblings.

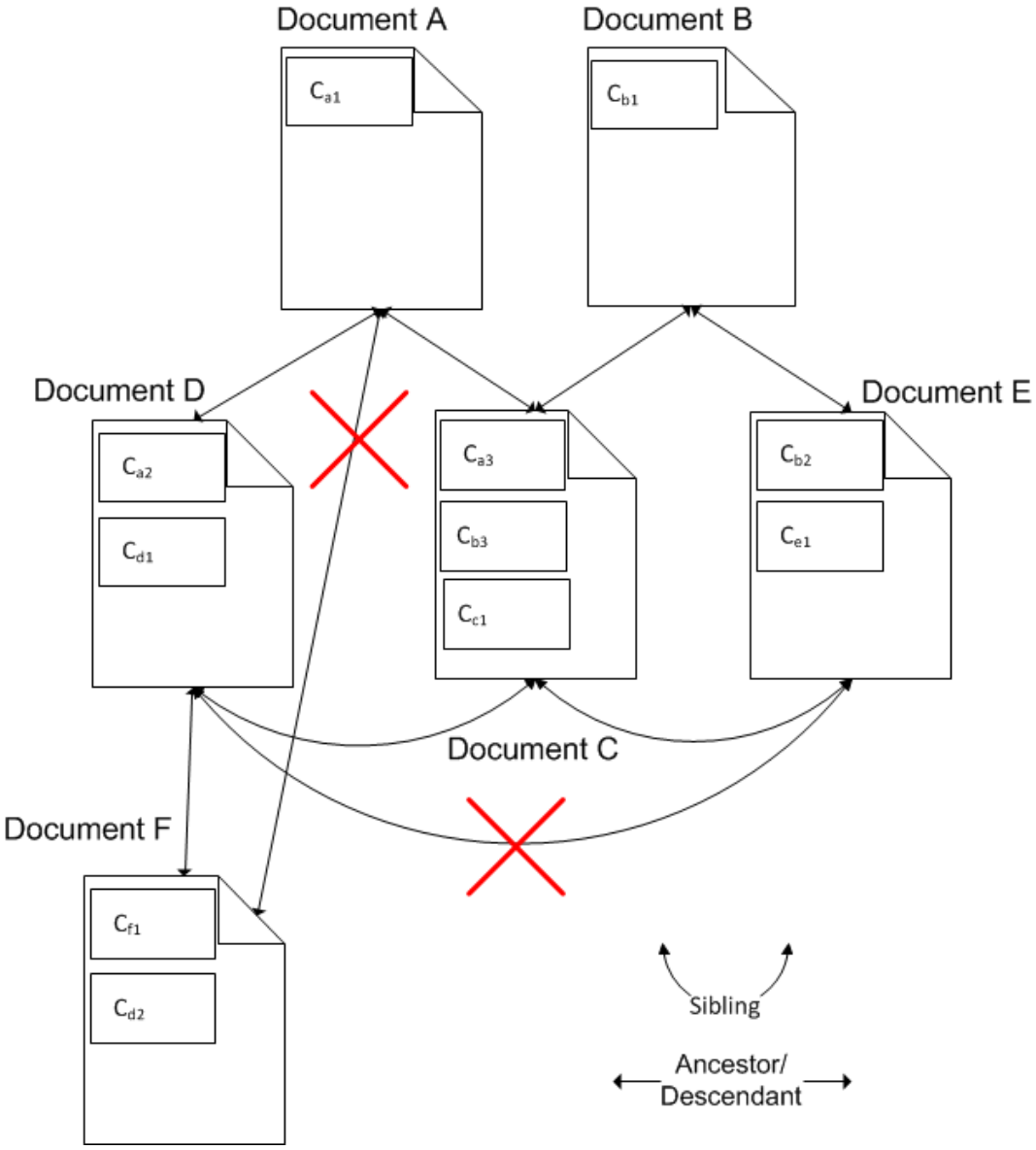


Figure 5.10: Non-Transitive Sibling Relation

5.2.7 Queries and Scenarios

This section gives an overview of possible queries that use the document relations, as well as scenarios for those queries. To keep the scenarios simple, we assume documents only include one piece of content. If a document would contain more than one piece of content, the relation to the other

documents would be displayed more than once, depending on the number of content pieces. For example, Documents A and B can both be siblings and ancestor/descendant at the same time, if they include pairs of content pieces with those relations.

Queries

We use the concept of queries to describe the information that can be derived from out DDNA model.

Definition 8 *(Query) A Query is a function* $Q : (\mathbb{D}, \mathbb{D}) \rightarrow \mathbb{D} \times \mathbb{E}$, $Q(B, S) = (R, E)$, *where* $\mathbb{D}$ *is the set of all documents and* $\mathbb{E}$ *is the set of all possible relations between the documents, and* $B, S, R \subset \mathbb{D}$. *B is called the base set and S the search set. The function compares the DDNA of each content piece of the documents in B and S and returns as query result a set of related documents* $R \subseteq (B \cup S)$ *and a set of relations E associated with the documents in R. R and E form a directed graph* $G = (V_R, E)$ *with the vertices* V_R *being the documents returned in R and the edges being the relations returned in E. The elements included in R and E depend on the actual query.*

The objects defined in Definition 8 have the following purpose:

- **Base set** B: The set of documents for which we seek relations.
- **Search set** S: The set of documents which are examined as possible relations to the documents from the base set.
- **Result graph** G: All documents from the base set and the identified documents from the search set that match the query, plus edges illustrating their relations.

Simple Queries

For simple queries the documents in base set B and search set S all contain single content pieces only. Table 5.1 includes the basic simple queries that are possible using the DDNA. The Set columns name the range for the cardinality for the base and search set with m standing for the maximum number of documents available to the user. The graph column names the range for the cardinality of the vertices of the result graph of R.

Query	Set B	Set S	V_R
Find all siblings	1,m	1,m	$\|B\|, \|B \cup S\|$
Find all ancestors	1,m	1,m	$\|B\|, \|B \cup S\|$
Find all descendants	1,m	1,m	$\|B\|, \|B \cup S\|$
Find all relations	1,m	1,m	$\|B\|, \|B \cup S\|$
Find oldest/youngest ancestor	1,1	1,m	$1, \|S\|$
Find oldest/youngest descendant	1,1	1,m	$1, \|S\|$
Find oldest/youngest sibling	1,1	1,m	$1, \|S\|$
Find shared ancestors	2,m	2,m	$\|B\|, \|B \cup S\|$
Find shared descendants	2,m	2,m	$\|B\|, \|B \cup S\|$
Find shared siblings	2,m	2,m	$\|B\|, \|B \cup S\|$

Table 5.1: Table of Queries

Simple Query Explained The function $q_s(B, S) = R$ is representing the "find all siblings" query. Therefore the following parameters are true:
$|B| \in [1, m]$,
$|S| \in [1, m]$, and
$|V_R| \in [|B|, |B \cup S|]$.
This means that the result graph vertices set contains at least the documents of the base set. This happens when no document from the search set is a sibling to a document in B. Documents from S will be included in V_R, if those documents are siblings to documents in B. This is a formal example of the definition, now some applied examples follow.

Simple Query Examples

In the example scenario, a user has several CV documents on their computer. We assume these documents to be the only available documents. This user can use the DDNA of the different documents and queries to explore the relations those documents share.

Example 1 — Find all Relations The result of the query "find all relations" with all documents as base and search set is illustrated in Figure 5.11. Since the user queried for all relations on all documents, the result graph includes all documents and relations.

Example 2 — Find all Ancestors/Descendants The query "find all ancestors/descendants" with all documents as base and search set returns the same graph as Example 1, without the sibling relations. All documents are included, as all documents are either ancestor or descendant of another

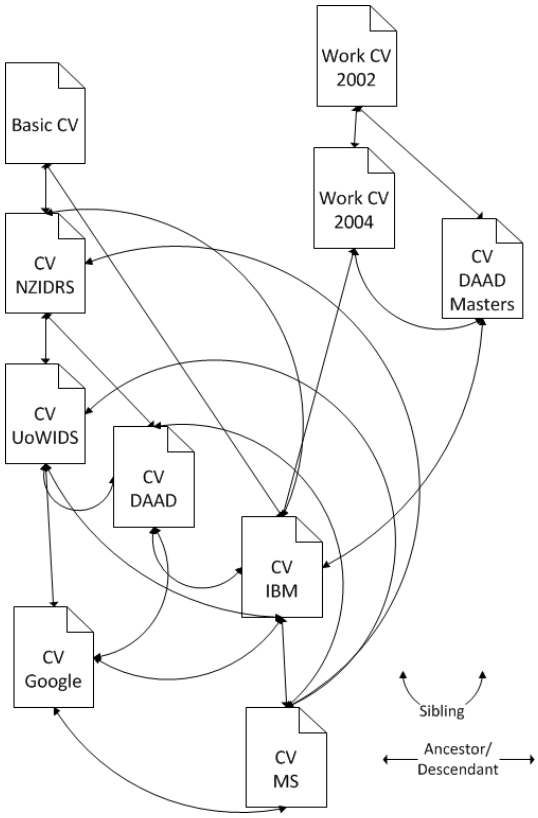


Figure 5.11: All Relations between the Documents

document.

Example 3 — Find all Siblings The query "find all siblings" with all documents as base and search set returns the same graph as Example 1, without the ancestors/descendant edges and without the documents `Basic CV` and `Work CV`. The two documents are excluded as they are not part of any sibling relationship.

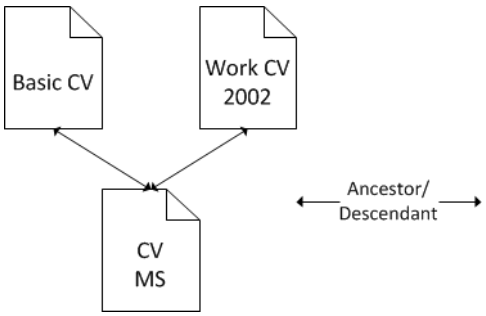


Figure 5.12: Oldest Ancestor

Example 4 — Find the oldest Ancestor Figure 5.12 illustrates the result graph for the query "find the oldest ancestor". The base set is the `CV MS`

document and the search set is constituted of all documents. The result graph includes the documents `Basic CV` and `Work CV 2002`, since both of these documents are ancestors of `CV MS`, but have no ancestor of their own. Note that all the other documents are missing since they did not have a relation to the *MS CV* document that matched the query.

Example 5 — Find shared Ancestors The result graph of the query "find shared ancestors" is illustrated in Figure 5.13. The base set is constituted of the documents `CV Google` and `CV MS` and the search set is constituted of all documents. The result graph includes the base documents `CV Google` and `CV MS` and the `Basic CV` document from the search set. The `Basic CV` document is the only document that is ancestor to both `CV Google` and `CV MS`. Again all other documents are not displayed, as they failed to qualify for the specified relation.

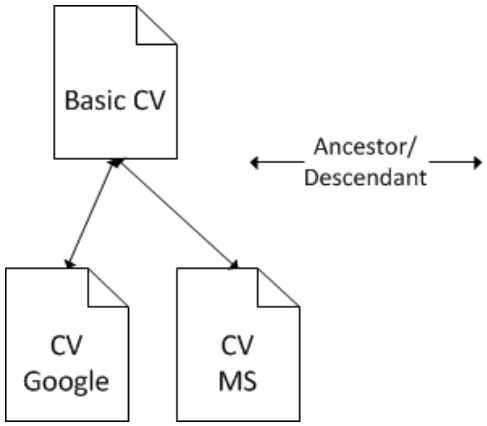


Figure 5.13: Shared Ancestor

Complex Queries

So far the examples have been simplified, since we only allowed one content piece per document. However, we defined that documents in base set B or search set S can hold more than one content piece, which allows for more complex queries. The following example assumes that a figure has been re-used via copy and paste and has been edited across a number of research papers.

Complex Query Example A user has found several papers that describe features and studies about the TIP system. They read an interesting paper about a study on a feature of the system, but the paper does not describe the feature or the system in full detail. However, the feature is illustrated

in a figure. The user now wants to know which other papers include this feature and most importantly which paper introduced it. They can do so by querying for all relations, limiting the query to DDNA related to the figure in the document.

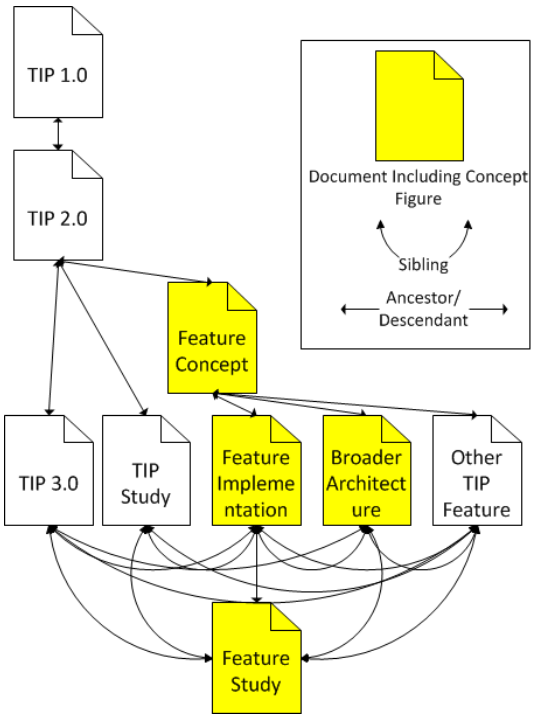


Figure 5.14: Papers and their Relations

Figure 5.14 shows the papers and the relations between them. The papers that actually include the figure the user saw are marked yellow. In this scenario, the user now queries for all documents that are related to the *Feature Study* document with the *Feature Study* document DNA reduced to the DDNA of the figure. The base set is the figure and the search set is composed of all papers available. The result graph is displayed in Figure 5.15.

Relation Strength — Concept Sketch

So far we defined the ancestor/descendant and sibling relations and illustrated some examples of how to use them in queries. However, these queries only reflected the type of the relation, not the strength of the relation. We here briefly sketch the concept of relation strength, which may be used as a measure for the similarity between two pieces of content, both semantically and syntactically.

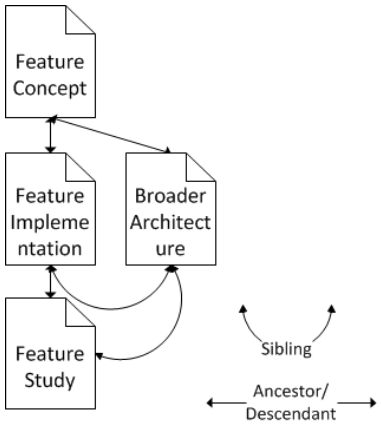


Figure 5.15: Result Graph for Figure Query

Example Content A and B are ancestor/descendant, but only share very little content due to heavy editing. Content C is also Ancestor of B but differs very little in content. In this example, the relation between C and B is semantically stronger than the relation between A and B.

5.3 Summary

In this chapter we addressed the question of how content-centered provenance data tracking could designed, so we can implement a system using that design. We created a model that defines the object of this research, documents and content. We also defined what actions can be performed on content and how these actions constitute sessions. We defined that sessions are used to mark changing points for the saved metadata, the DDNA. Following we showed what relations are detectable using DDNA and what queries could be performed knowing these relations.

Our system relies on detecting actions performed on content as they happen. We consider the third research question (How can content-centered provenance data tracking be implemented?) to be answered with the introduction of this model. The fourth requirement (Automated creation of the metadata) is dependent on the implementation of the design, which is described in detail in the next chapter. Our model fulfills Requirements 1–3 (Relatedness, Relationship Metric and Distributed).

Relatedness The DDNA is always kept and modified when manipulating content and transferred with copied content. As a result, we can always

determine if two pieces of content are related by comparing their DDNA.

Relationship Metric We defined different relations content can be in, such as ancestor/descendant and sibling relations. These different types of relations fulfill the metric requirement.

Distributed The DDNA is stored within the same document as the content, and is also transferred when the content is copied to another document. As a result, DDNA cannot be disconnected from the content, which fulfills the third requirement.

6

Implementation

Chapter 5, introduced the DDNA model, which fulfills the requirements for a distributed content-centered provenance data tracking system. In this chapter, we report on a software prototype implementing the model. The prototype will be used to answer the fifth research question and provides us with the means to measure the increase in result quality. We explain which parts of the model are represented by which parts of the implementation and how these parts fulfill the requirements set out in Section 2.4.4.

We start this chapter by specifying the software tools used, based on the result of the studies in Chapter 4 and present a detailed walk-through and architecture of our prototype. Next, we introduce the DDNA Tracker and DDNA Analyzer. The DDNA Tracker is an implementation of the model proposed in the last chapter. The DDNA Analyzer can analyze the DDNA of different pieces of content and determine their relationship and also provides a user interface to utilize that information. The DDNA Analyzer is needed to conduct user studies to verify the use of provenance data for knowledge workers. Finally we present a second in-detail walk-through which shows how the prototype functions following one example. We finish by summarizing how the software prototype matches the DDNA model.

6.1 Software Used

We now specify the software we used. To ease the introduction of our prototype, we needed to disrupt the work environment of knowledge workers as little as possible. Therefore, we implemented our prototype using software the knowledge workers were already using. This minimized time spent on learning to use the prototype and the disruption caused by it. As a result, the following two guidelines need to be followed when choosing the software to be used to implement the prototype:

1. The software needed to be used by a large percentage of knowledge workers.
2. The software needed to allow for implementation of our prototype.

Following these guidelines we chose Microsoft Word (Microsoft, 2014) to be the host for the DDNA Tracker and DDNA Analyzer. We implemented both add-ins using Visual Studio and C#. The DDNA Analyzer also uses the QuickGraph (pelikhan (2011)) library to create a graph representing the relations between different documents and the GraphX library (PantheR (2014)) for the visualization of the created graph. Both libraries are open source and freely available under the public domain.

Microsoft Word The studies in Chapter 4 identified Microsoft Word as the document editor most used by knowledge workers, making it the best choice for implementing a provenance data tracking system. Microsoft Word allows for add-ins to be installed, which is the least intrusive way of introducing the prototype into the work environment of the knowledge worker. Since Microsoft Word differs slightly depending on the operating system, we chose to support Microsoft Word for Microsoft Windows 7 (Microsoft, 2011).

C# By using Visual Studio and C# to implement the prototype we adhered to the second guideline. Along with Visual Basic C# is one of Microsoft's the most supported programing languages in relation to Microsoft Office customization. Furthermore, Microsoft provides a large support knowledge base, providing information on how to access APIs and build Microsoft Office customization add-ins using Visual Basic and C# code examples.

Alternatives We also considered using an open source document editor like Open Office (Oracle Corporation, 2011) or a specialized editor like LaTeX (Lamport, 2011). Since the code base is easier to access in these identified alternatives, the development process would be easier. However, such specialized document editors would mean a limited user base available for studies. This would make conducting studies measuring the usefulness of the DDNA more difficult.

6.2 Architecture and Walk-through

The prototype is constituted of two parts, the DDNA Tracker and Analyzer. The high level architecture of our prototype is shown in Figure 6.1.

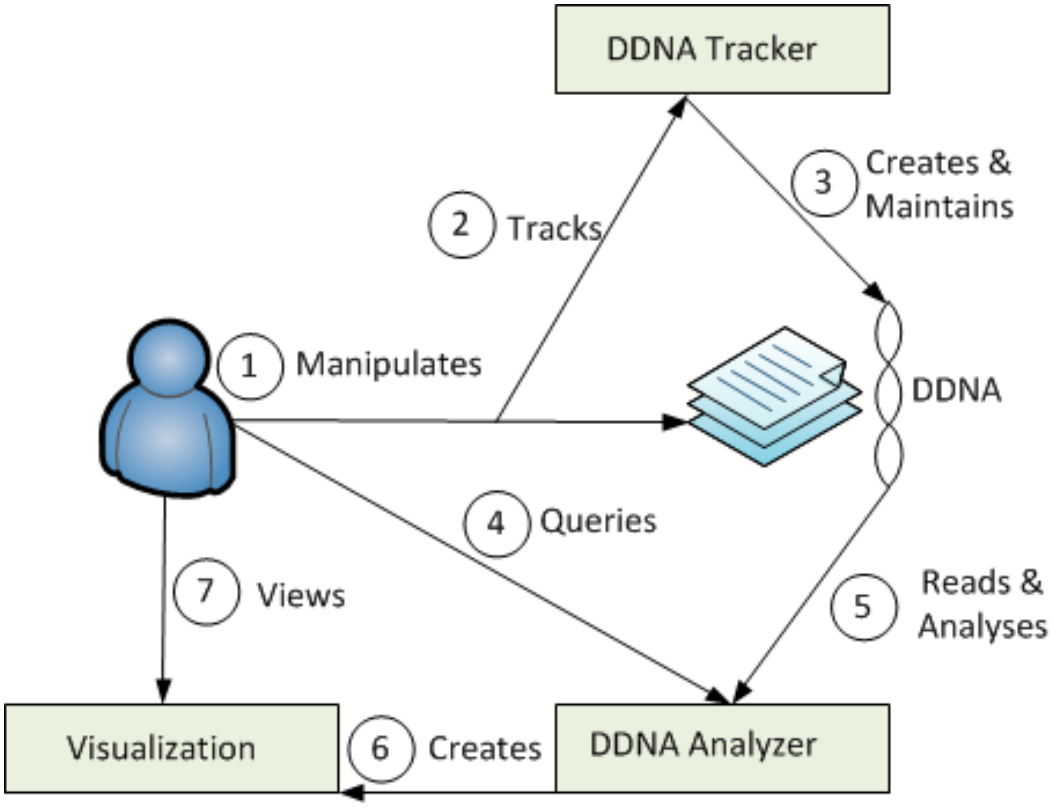


Figure 6.1: High Level Architecture and User Interaction

Step 1 & Step 2 — Tracking of User Manipulation As soon as a document is opened, the DDNA Tracker tracks every manipulation executed on that document via Microsoft Word. This includes copy and paste commands between documents. Tracking manipulations are stopped when a document is closed.

Step 3 — DDNA Creation and Maintenance The DDNA Tracker creates and maintains the DDNA for each document that manipulations are tracked for. If a document without a DDNA is opened, or a new document is created, the DDNA Tracker creates a new DDNA for the document and starts maintaining it. In all other cases, the existing DDNA is maintained. The DDNA is directly attached to the document and is therefore saved with the document when the document is saved.

Step 4 & Step 5 — Query and Analysis The user can query the DDNA Analyzer about the relationships between accessible documents. The DDNA Analyzer reads the DDNA of all accessible documents and analyses the implicated relations.

Step 6 & Step 7 — Visualization As a result of an analysis, the DDNA Analyzer creates a visualization of the relationships appropriate to the query and returns this visualization for the user to view.

The next two sections introduce the DDNA Tracker and Analyzer in detail, whilst linking back to these six basic steps.

6.3 DDNA Tracker

The DDNA Tracker is a Microsoft Word add-in that implements the model introduced in Chapter 5. Our implementation is built to track and process the evolution of content used in Microsoft Word documents and store this information with the document. The aim of the DDNA Tracker is to be as un-intrusive as possible, while simultaneously being as accurate as possible.

Due to DDNA Tracker being a Microsoft Word add-in, it starts and closes with every instance of Microsoft Word, therefore never missing any changes made to content inside MS Word documents. The general architecture of the DDNA Tracker is shown in Figure 6.2. The DDNA Tracker is embedded into Microsoft Word and needs to run in Microsoft Windows 7 or higher in order to function. The interactions of each of the parts are described in the following subsections, which also describe the functions of each part in detail. The directions of the arrows show the flow of information between the different parts.

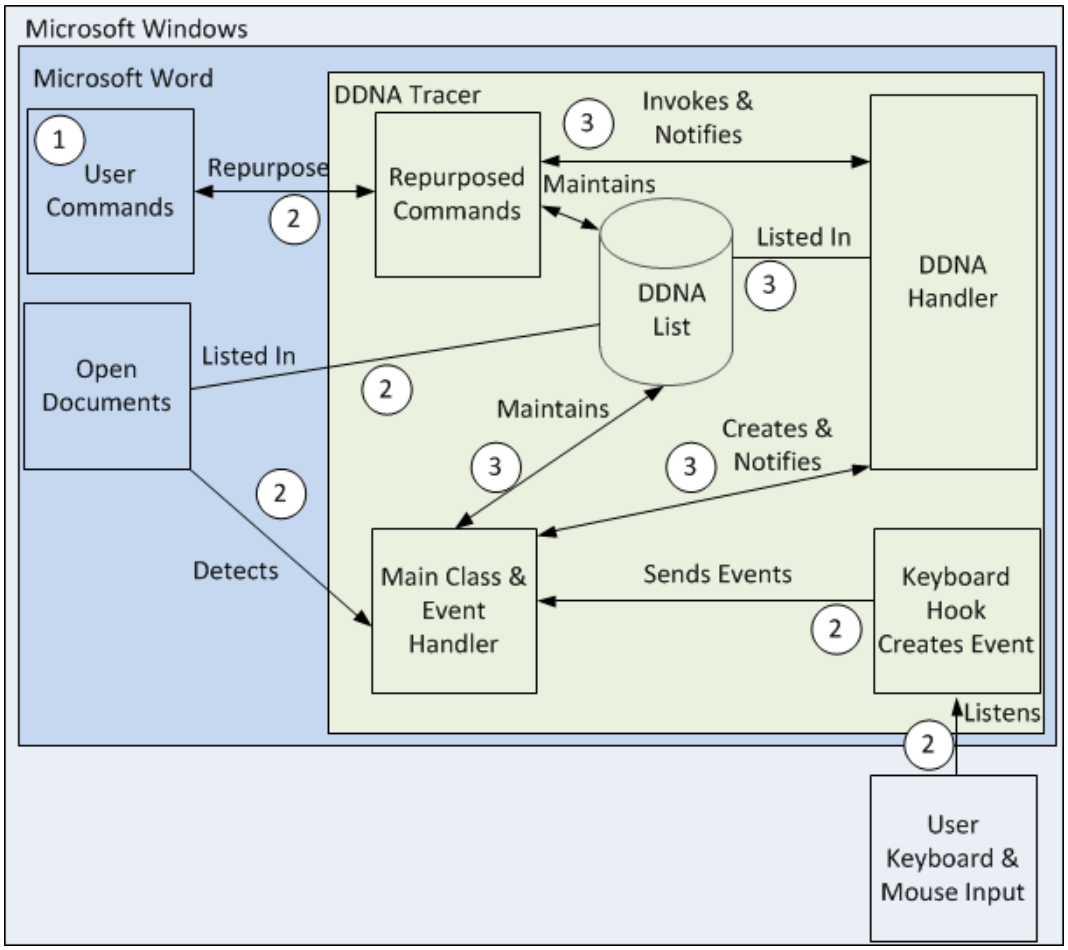


Figure 6.2: The Software Architecture of the DDNA Tracker. The circled numbers refer to steps mentioned in Figure 6.1.

6.3.1 Main Class

The Main Class serves two functions. The first function is to listen to events created by Microsoft Word for documents opened, closed, and saved. The second function is to listen to events created by the Keyboard Hook module. These events represent inputs made by the user (Step 2). These events are used to notify the DDNA Handler of changes made to the content.

Document Open When a new document is opened, the DDNA Tracker creates a new DDNA Handler object for the document and creates an entry in the global DDNA List.

Document Save If a document is saved, the DDNA Tracker saves a copy of the previous version of the document into a local DDNA Archive folder and then notifies the DDNA Handler associated with the document about

the save command being invoked. The handler returns an updated DDNA which the old DDNA is replaced with.

Document Close If a document is closed, the DDNA Tracker removes the entry for the responding document from the global DDNA List.

User Input on Document When this event is triggered, the DDNA Tracker verifies which document is currently active and notifies the DDNA Handler associated with this document of the change made to the document.

DDNA Model The document open, save and close events are used to identify Sessions as introduced in our model. The user input event is used for detecting Actions as identified in our model. Saving old versions of documents into the DDNA Archive reflects the policy that each save leads into a new document.

6.3.2 Keyboard Hooks

This module creates low level hooks, which notify the DDNA Tracker when a key on the keyboard is pressed. The module evaluates if the combination of keys pressed can result in content changes in a word document and whether or not Microsoft Word is the active window at the time of the event. If both conditions are true, an event including the pressed key is created and sent (Step 2 in Figure 6.1).

DDNA Model The events created are representations of the Action concept introduced in our model. Each event represents one Action.

Example 1 The user presses the 'A' key, but the active window is returned to be a browser, no event is triggered.

Example 2 The user presses the 'A' key and the active window is returned to be Microsoft Word. An event is triggered.

Example 3 The user presses the ctrl key and the active window is returned to be Microsoft Word. No event is triggered.

6.3.3 Re-purposed Commands

We re-purposed the three commands copy, cut and paste as these are the three main commands offered by word to directly manipulate content (Step 2). When implementing a Microsoft Word add-in, it is possible to

re-purpose commands available in Word. This means that either the code behind these commands can be replaced with code from the add-in, or additional code can be added to the command.

DDNA Model The re-purposed commands represent the Actions of the same name introduced in our model.

Copy

The copy command has been replaced with our own copy command. When selected content is copied, the re-purposed command triggers the following four steps:

First Step The DDNA Handler is called to process all current changes stored in regards to the selected document and update the DDNA accordingly.

Second Step The DDNA Handler is called to return the DDNA associated with the selected content. That DDNA is pasted as a string in front of the selected content and the selection to be copied is extended to include that string.

Third Step The selection is copied, which means it is added to the clipboard.

Fourth Step As the final step, the DDNA is removed from the selected content.

These steps happen too fast for the user to notice. The DDNA string is visible in the clipboard, attached to the content as shown in Figure 6.3.

Cut

The cut command has been replaced with an invocation of our re-purposed copy command followed by a fifth step.

Fifth Step Remove the content selection and notify The DDNA Handler of the content change to process the DDNA(s) appropriately.

Paste

The paste command has been replaced with our re-purposed paste command. When content is pasted, the re-purposed command triggers the following four steps:

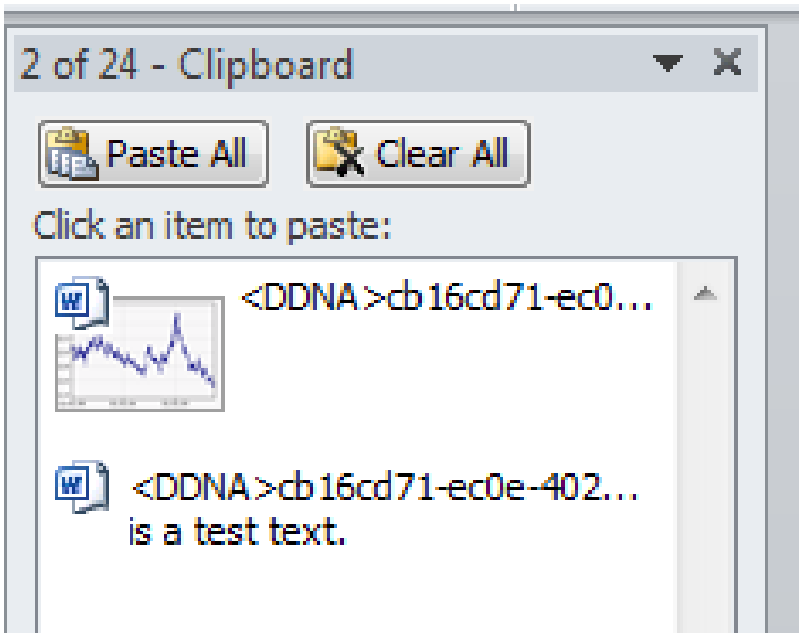


Figure 6.3: The DDNA Inside the Clipboard

First Step The paste command first executes paste as implemented in Microsoft Word.

Second Step The pasted selection is scanned for a DDNA string. If a DDNA string is detected, it is sent to the DDNA Handler along with the character positions of the start and the end of the range, to be included into the DDNAs of the document.

Third Step The DDNA string is removed from the pasted content.

Fourth Step The DDNA Handler is notified of the content manipulation at the selected range and adds it to the current list of changes made to the content.

6.3.4 DDNA Handler

The DDNA Handler module is used to manage all DDNAs of all opened documents (Step 3). Each handler instance represents one document. A handler is invoked with a newly opened document and can be used to retrieve a DDNA associated with specific content. It can also be used to track changes to content and update the DDNA of that content. The handler is also called when content is pasted into a document with a DDNA attached. The attached DDNA is then sent to the handler to be included in

the DDNAs of the target document.

DDNA Model The DDNA Handler implements the process of concentrating the content history Z into the DDNA, as proposed in our model.

Invoke

The DDNA Handler needs a docx document to be invoked. When created, the handler verifies if the document has DDNA information included in its custom XML part (part of the .docx structure). If DDNA(s) are detected, they are extracted and added to the instance of the handler as a list. If not, a new DDNA is created and added to the DDNA list. This DDNA has a new UUID and the range starts at character position 1 and ends at the last character position in the document. One tick is added to ticks, representing the time that this DDNA was created. Ticks are processor time representations.

Track Changes

Changes are submitted to the DDNA Handler as a set of two integers, the starting character position of the change and the last character position of the document after the change.

Every time the DDNA Handler is notified of a change made to the document, that change is stored into a change list. Each entry contains the starting character position p of the change and nature n of the change. n is determined by calculating the difference of the position of the last character in the document before and after the change, for example +1 or -1. For example, if a user enters the character 'A' at position 50, $p = 50$ and $n + 1$, but if the user deletes the word 'The' starting at position 1, $p = 1$ and $n = -4$.

Update DDNA

The update function of the module is called internally for two reasons. The first reason is if the handler is called to return DDNA of content in the represented document. The second reason is if the handler is called to add a new DDNA to the DDNAs representing the document. To update all listed DDNAs, the handler processes all changes stored in the change list. All changes with $n \neq 0$ lead to adjusted range sizes for the different DDNA(s), depending on p. We identify the start position of a range as rs

and end position as re.

$p > re > rs$ If n is positive, no changes are made to the range as the contents of the range were not affected. If $n < 0$, we calculate if $p - n$ falls into the range. If that is the case, the new end position of the range is set to $re = min(p - n, rs)$.

$p < rs < re$ The change to the content affects the ranges before this one. Therefore both the start and the end position of the range are adjusted to $rs = rs + n$ and $re = re + n$.

$rs \leq p \geq re$ The end position of the range is adjusted to $re + re + n$. If n is negative, the new end position of the range can only be reduced to the start position of the range.

After all changes are processed, the handler also adds a new tick to the ticks of the DDNA including changed ranges. Ranges with the same start and end position are removed.

Add DDNA

When content is pasted to a document, the handler is called to add the DDNA of the pasted content to the DDNA list of the target document. The handler is called with the new DDNA and the start and end position of the pasted content. After the handler has processed all changes up until the paste happened, it processes another change with p being the start position of the paste and n being the total character size of the pasted content. However, the range which the paste happened in is processed differently.

$rs \leq p \geq re$ This range is split into two ranges R_1 and R_2. The new end position re_1 of R_1 is set to $re = p - 1$. The new start position rs_2 of R_2 is set to $rs_2 = p + n$ and the new end position re_2 of R_2 is set to $re_2 = n + re$.

Retrieve DDNA

When called to provide the DDNA of a part of the document, the DDNA Handler first processes all the changes stored (see update). The updated DDNA is then returned. The new DDNA is not written into the custom XML part of the document, but only held as part of the DDNA Handler.

6.3.5 Custom XML Parts

The .docx format for Microsoft Word documents allows for custom XML strings to be stored directly into the document structure, using the custom XML part of the format. There is an API for extracting and deploying custom XML parts in .docx documents. We use the custom XML part to store the DDNA (Step 3). The DDNA for our prototype has four parts:

1. A UUID to uniquely identify a piece of content. This UUID is static and never changes.
2. A series of time stamps in the form of DateTime.Ticks supplied by C#. This series is augmented with every save as a new tick is added.
3. The range of the content. The range contains the first character's position and the last character's position of the content. A piece of content can be distributed over several ranges.

The three parts are put together in this format:

```
<DDNA>
 <UUID>uuid</UUID>
 <Ticks>ticks</Ticks>
 <Ranges>
  <Range>range</Range>
 </Ranges>
</DDNA>
```

DDNA Model The DDNA is representing the accumulation of the history Z introduced in our model.

UUID & Ticks

The UUID is a unique identifier assigned to a piece of content. Two pieces of content are related if they share the same UUID. The relationship between related content pieces can be further specified by comparing the ticks of both content pieces. If the ticks of Content A are a true subset of Content B, then A is ancestor of B and B is descendant of A. Content A and B are siblings, if their ticks include an identical true subset. The shared ancestor of A and B is the content piece with that subset as ticks and the same UUID as A and B.

The Ticks are also used to establish a quantitative measure for distance between two pieces of content with the same UUID. The greater the number of ticks of two related content pieces differ, the more versions lie between them.

DDNA Model The UUID & Ticks are used to identify the Ancestor, Descendant and Sibling relations introduced in our model. The Ticks are also used to give a measurement of relation strength, as introduced in our model.

Range

The ranges assigned to a piece of content allocate which content in a document is associated to which DDNA stored for that document.

DDNA Model We introduced C as part of the document concept in the last chapter. Each document is defined in order to hold different content pieces $C_1 - C_n$. Ranges are used to identify the position of a content piece inside the document.

6.3.6 Requirements

We introduced the requirements for our system in Section 2.4.4. The DDNA Tracker fulfills all of the requirements.

Relationship Detection Relationships are easy to detect, since the DDNA is always attached to copy and pasted content and integrated into the target document. Two DDNAs sharing the same UUID are related. Only content that is copied manually via typing does not result in a relationship formed.

Relationship Metric The Ticks part of the DDNA allows for quantitative measurement of the strength of the relation between two pieces of content. The more two Ticks strings of two pieces of content differ, the less related they are.

Distributed The DDNA Tracker stores the DDNA inside the document including the associated content, therefore fulfilling the distributed requirement.

Automated The DDNA Tracker works without any user input and therefore fulfills the requirement of being automated.

The next section introduces the DDNA Analyzer. This part of our prototype enables users to query the data collected by the DDNA Tracker, as the DDNA Analyzer is built to analyze all available documents' DDNA and create a visualization of the relationships detected as a result of the analysis.

6.4 DDNA Analyzer

The DDNA Analyzer is a Microsoft Word add-in that is built to extract the DDNA attached to documents and provide a comparison of different sets of DDNA. It also provides an interface to visualize the relationships deducted from comparing the DDNAs. The architecture of the DDNA Analyzer is shown in Figure 6.4.

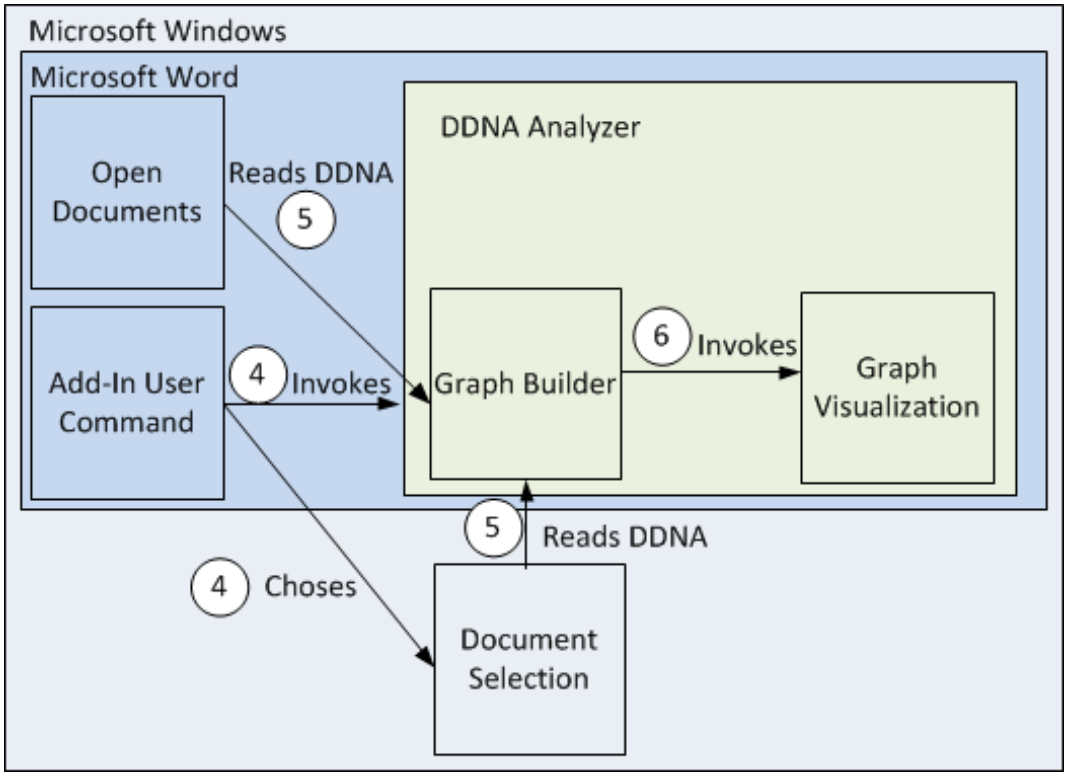


Figure 6.4: The Software Architecture of the DDNA Analyzer. The circled numbers refer to steps mentioned in Figure 6.1.

6.4.1 Graph Building

The Graph Building module's aim is to create a directed graph representing the relationships between documents (Step 5). The graph includes vertex

and edge objects. Vertices either represent a document and all contained content pieces, or a content piece with no document found including it.

DDNA Model The edges between vertices represent the ancestor & descendant relationship, with the source vertex being the ancestor and the target vertex being the descendant. Sibling relationships are not directly displayed, but can be deducted from the ancestor & descendant relationships.

Vertex A vertex object in the created graph must be associated with at least one UUID and Ticks combination, but can be associated with more than one combination. A vertex can be associated with a document path. The document found in that path must include a DDNA for all UUID and Ticks combinations represented by the vertex.

Edge An edge object must always be associated with an UUID, a target vertex and a source vertex. The source and target vertex must represent an UUID and Ticks combination that has the same UUID as the edge. The Ticks of these two combinations must only be one time stamp apart, with the source having one time stamp less than the target.

Commands The Graph Building module is called by the two commands *Compare Selection* and *Scan Complete Folder*. The only difference between the two commands is the base set of DDNAs to start the comparison and the target set of documents to compare with. Both commands call the Graph Visualization module once analysis is complete.

Command: *Compare Selection*

The *Compare Selection* command results in the DDNAs associated with the content selection made by the user to be extracted and compares them to all DDNAs included in the documents contained in the folder chosen by the user.

The DDNA Analyzer first processes the information found in the starting document.

Step 1 The Analyzer accesses the custom XML part of the opened document to read out the DDNAs associated with the selected content. This is done by selecting the DDNAs that include ranges that overlap with the selected range. Each DDNA's UUID is stored in a list to serve as the base

set.

Step 2 The first vertex of a graph depicting the relations between content pieces is created. This vertex is associated with all UUIDs and the associated Ticks strings from the opened document. The vertex is also associated with the physical document location.

Step 3 For each UUID found, the latest tick is removed from the Ticks string. If the Ticks string is not empty, a new vertex representing that UUID and Ticks combination is created. The DDNA Analyzer also creates a new edge. That edge has new vertex as source vertex and the vertex representing the UUID and unmodified Ticks as target vertex. The Analyzer repeats this step until the Tick string has only one time stamp left. These vertices are not associated with documents, but instead are marked as missing documents.

The DDNA Analyzer is now finished analyzing the base set and continues by analyzing the search set.

Step 4 The DDNA Analyzer now searches the target folder for .docx files and tries to extract DDNAs from these files. If a DDNA XML is found, the UUID of that XML is compared to the UUIDs in the base set list. If the UUID is included in the base set list, the DDNA Analyzer compares the UUID and Ticks combination of that DDNA with the combinations of all already created vertices.

Step 5a — If a match is found If a match is found, and the vertex is not associated with a document, the vertex is adapted to be associated with the document the current UUID and Ticks combination is read from. If the document is already represented by a another vertex, the two vertices are merged. In the rare case that the UUID and Ticks combination is already associated with another document and vertex, the analyzer creates a parallel vertex duplicating all the edges of the other found vertex. This can happen when users keep duplicate files in different locations.

Step 5b — If no match is found The DDNA Analyzer checks if the document the UUID and Ticks combination is taken from is already represented by another vertex. If no vertex is representing the document, a new vertex is created and Steps 2 and 3 are repeated for that vertex. If that is the case, new vertices and edges are created removing the last time stamp of the

Ticks string until either a matching vertex is found, or the Ticks string has only one time stamp.

Steps 4 and 5 are repeated with the folder containing the archived documents (saved by the DDNA Tracker). The vertices representing an archived document are marked accordingly. Archived documents are only included in the graph if a vertex exists representing their UUID and Ticks combination that is not associated with a document.

Command: *Scan Complete Folder*

The *Scan Complete Folder* results in a complete scan and analysis of all DDNAs associated with any document in a chosen folder.

This command results in the same graph building steps as the *Compare Selection* command, but without creating a base set list of DDNAs. Instead, any DDNA that is found is incorporated into the graph that is build. This can result in multiple disconnected graphs if the folder contains documents that share no related content.

6.4.2 Graph Visualization — Interface

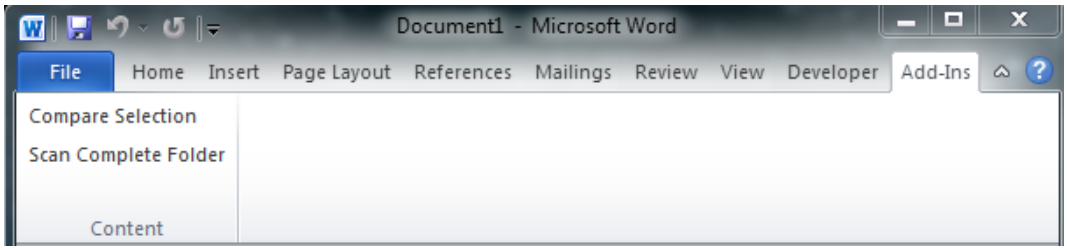


Figure 6.5: Interface of the DDNA Analyzer inside Microsoft Word

The DDNA Analyzer is a Microsoft Word add-in and can be used via the `Add-Ins` tab as shown in Figure 6.5. Its purpose is to create a visualization of the graph built by the Graph Building module (Step 6).

***Command:* Compare Selection** The *Compare Selection* command extracts the DDNAs related to the selected content. The user is asked to specify a folder. The documents in the chosen folder and sub-folders are scanned for DDNAs related to the extracted DDNAs.

***Command:* Scan Complete Folder** The *Scan Complete Folder* command will prompt the user to select a folder and then compares all pieces of

content in all documents contained in the folder and sub folders.

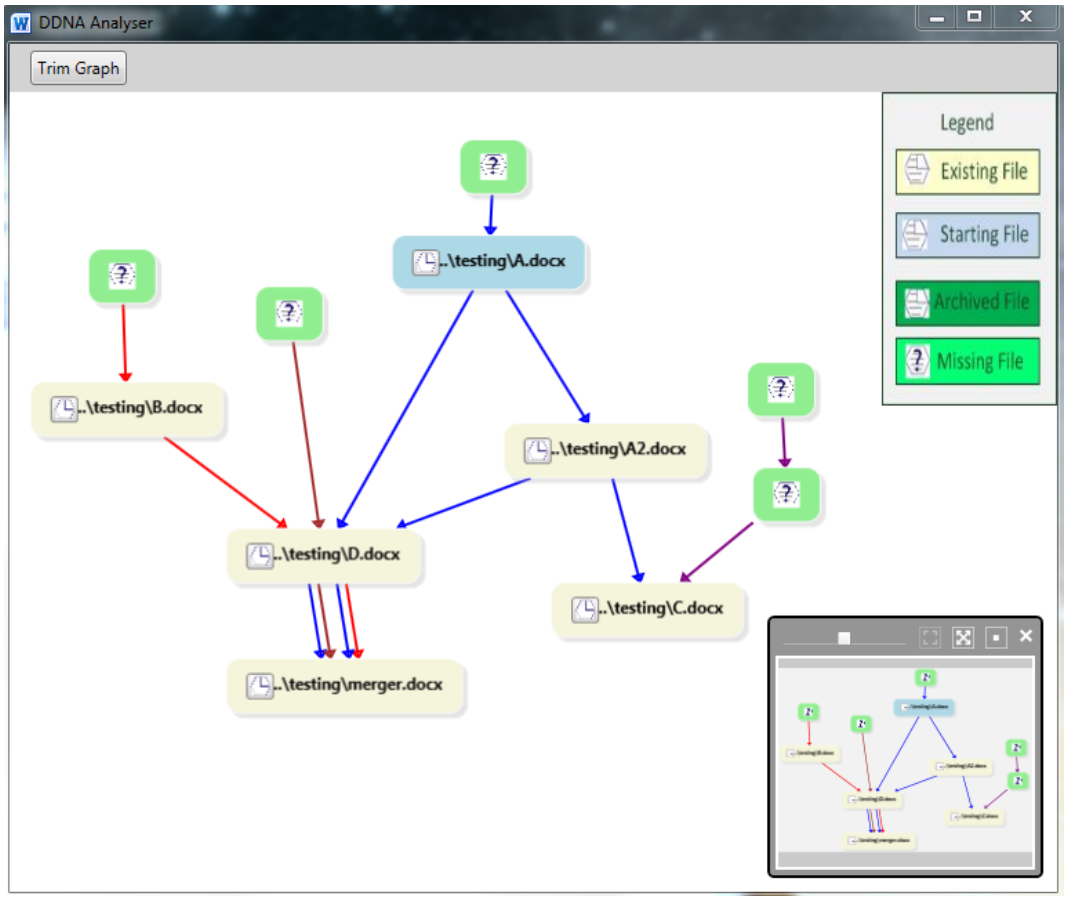


Figure 6.6: Visualized Graph

The Graph Visualization module uses the GraphX library to create a WPF (Windows Presentation Foundation) window displaying the graph. The shape of the graph is built using a layered graph drawing, also known as Sugiyama-style graph drawing (Sugiyama and Misue, 1991), algorithm supplied by the GraphX library. An example of this interface is shown in Figure 6.6.

Vertices Vertices are visualized as rectangular boxes that are either blue, beige, dark green or light green. Blue represents the document the Analyzer was called from and beige represents all other found documents. Light green vertices have no document associated to them and dark green vertices have documents found in the archive created by the DDNA Tracker. The box representing a vertex displays a shortened file path that can be expanded by hovering over the vertex. The box also includes a button to open the document represented by the vertex.

Edges Edges are visualized by arrows of different colors. Each color represents a single UUID, allowing the user to trace each content piece individually. Hovering the cursor over an edge results in the time difference between two vertices being displayed. This is the time difference between the last tick of the source UUID and the last tick of the target UUID.

Map The interface also includes a map for navigation of bigger graphs. Clicking on a part of the graph inside the map centers the main window on that part of the graph.

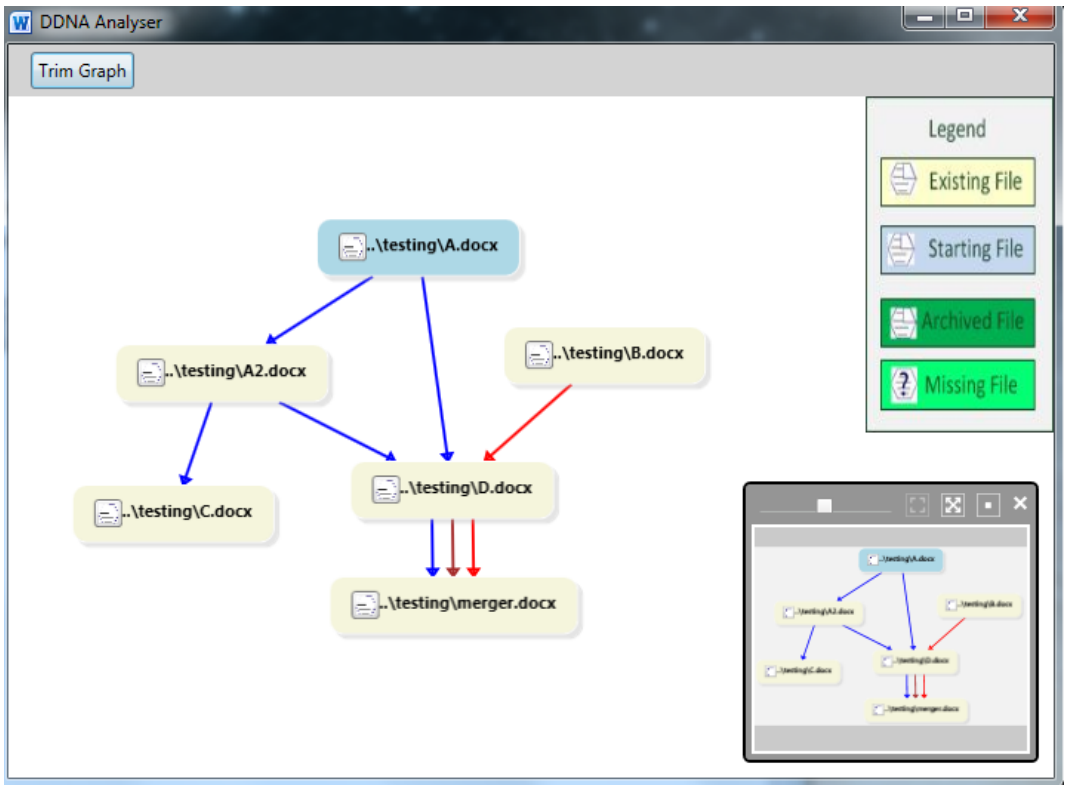


Figure 6.7: Visualized Trimmed Graph

The interface also provides the option to remove any vertex that does not represent a document (light green and dark green), therefore removing versions from the graph that the user cannot access. Relations between the other vertices are preserved. The result of trimming the graph shown in Figure 6.6 is shown in Figure 6.7.

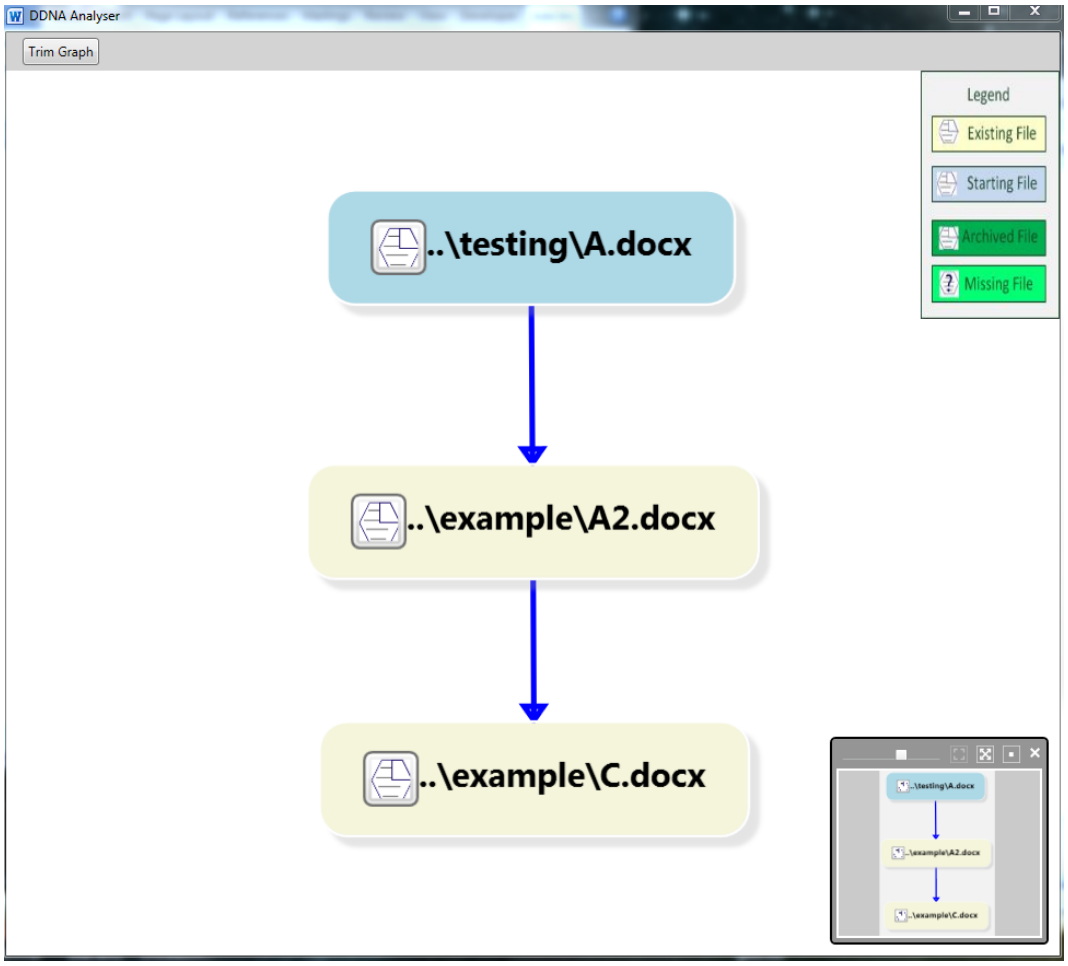


Figure 6.8: Example Case: Steps one to three

6.5 Detailed Walk-Through

This section introduces a detailed walk-through including five manipulated documents. This example includes five different Word Documents, A, A2, B, C and D. This example details the creation and re-use of content across different documents and describes the resulting relationships between them.

First Step We created Document A and inserted the content: "Some testing text.". A DDNA was created as soon as the document was created, but was not yet saved to the custom XML part of the document. The document was then saved, at which point a unique DDNA XML was added to the custom XML part of Document A.

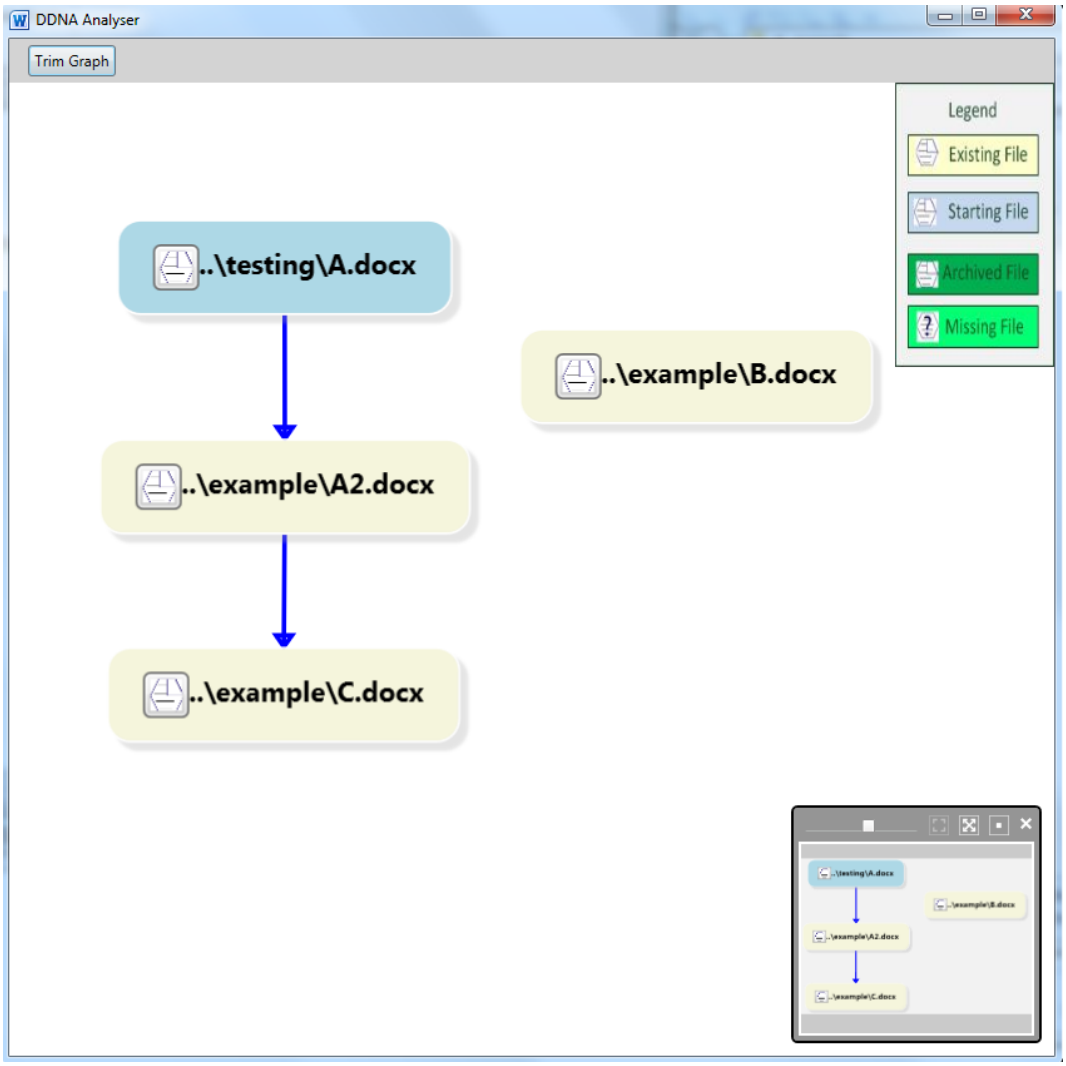


Figure 6.9: Example Case: Step Four

Second Step We added the content line "This is version 2." to the document and saved it as Document A2. The DDNA attached to the document was adapted by increasing the range of the content and adding a time stamp to the Ticks part of the DDNA.

Third Step We created Document C and copied and pasted the contents of A2 into C. Before we saved Document C, we replaced the content with the line "This is C." and saved the document. When C was created, a unique DDNA was created for the range 0 to 1. Since we copied and pasted content from A2 into C, the DDNA for that content was added to C. C now had two DDNA XMLs included in its custom XML part. When we analyzed the folder containing Documents A, A2 and C, we obtained the result shown in

Figure 6.8. The blue edges represented the path the content of Document A took. The original DDNA of Document C was not represented, as it had not evolved or was passed on to other documents.

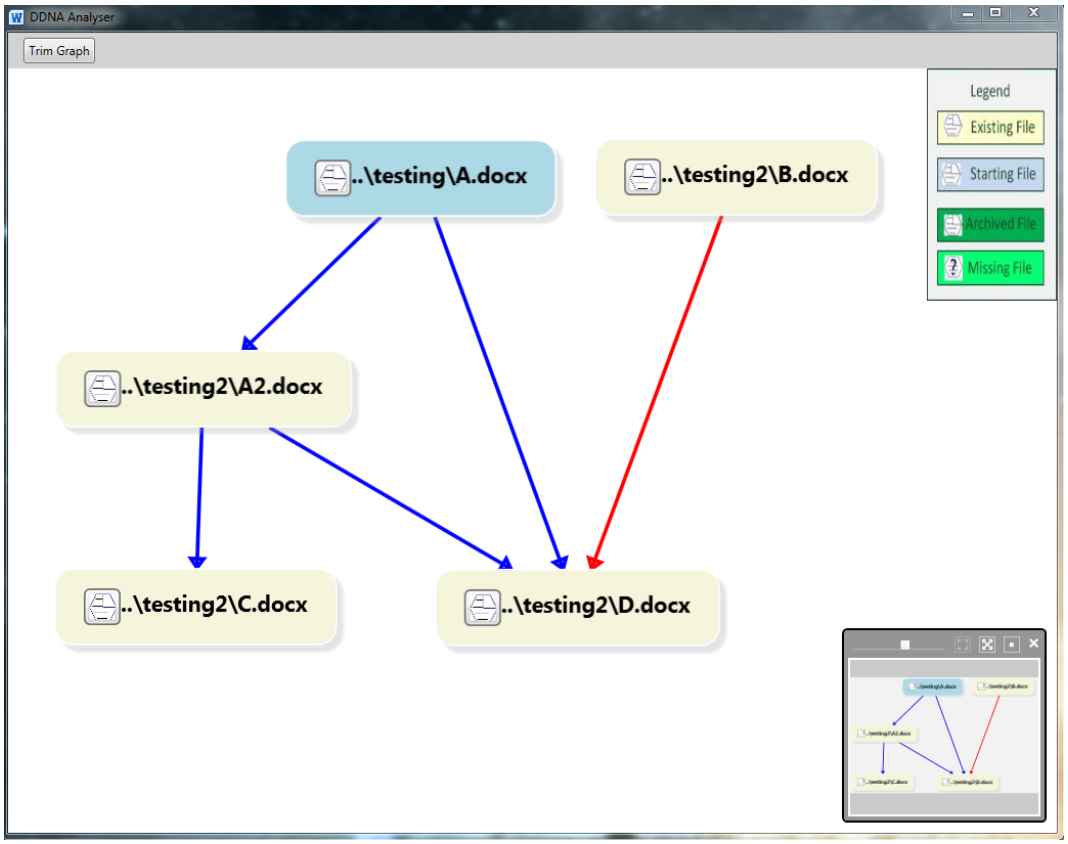


Figure 6.10: Example Case: Step Five

Fourth Step We created a new document and added the content: "This is B.". We now saved the content as Document B. Analyzing the folder containing Documents A, A2, C and B gave the result shown in Figure 6.9. B is not shown to have any connection to Documents A, A2 and C, since no content has been shared between the documents.

Fifth Step We created another new document and copied and pasted the line "Some testing text." from A, the line "This is version 2." from A2 and the line "This is B." from B into it, and saved the document as Document D. Document D now contained four different DDNA XMLs in its custom XML part: Document A's, A2's and B's with an added time stamp and adjusted ranges each and the DDNA created when creating Document D.

Scan Complete Folder A full folder analysis provided the result shown in Figure 6.10. There are two blue edges representing the transfer of content

from A and A2 to D and a red edge representing the transfer of content from B to D. Since the contents from A and A2 share an ancestor, they share the same color. The content from B is not related, which is why it has its own color, red.

Compare Selection Using the *compare selection* command on content of A or A2 resulted in the Documents A, A2, C and D being included in the resulting graph. This is because the selected content is included in these documents. If we use the same command on B's content, only Documents B and D are contained in the result. Selecting all content from D and running the command yields the same result as the full folder analysis, since D contains content related to all other documents.

6.6 Summary

In this chapter, we presented the DDNA prototype system consisting of the DDNA Tracker and DDNA Analyzer. The DDNA Tracker fulfills the requirements set in Chapter 2 by implementing the DDNA model introduced in Chapter 5. The DDNA Analyzer provides to the user an interface to query and visualize the gained information.

The DDNA Analyzer creates a directed graph visualizing the ancestor and descendant relations between documents as edges. Sibling relations can be detected by the edge color, since the color represents a unique starting point for content. If two vertices have an incoming edge sharing the same color but no ancestor descendant relation, they are siblings.

Users can detect the newest or oldest version of content by following the graph to its roots or leafs. Since the DDNA is stored according to the content included in a document, the user is able to execute the complex queries described in the last chapter by selecting a distinct piece of content in a document and executing the analysis on that piece of content.

While there is potential for future improvements, the DDNA Tracker and Analyzer are sufficient for studying the implications of provenance data. In the next chapter, we use this prototype to answer the final research question: Does content-centered provenance data tracking increase the result quality of the tasks identified?

7

Evaluation — In-Lab User Study

In Chapters 5 and 6, we introduced the DDNA model and its prototypical implementation of a content-centered provenance data tracking system, respectively. Using the DDNA prototype, this chapter answers the 4th and 5th research questions:

- How can we measure the effect of using content-centered provenance data tracking?
- Does content-centered provenance data tracking increase the result quality of the tasks identified?

We answer Question 4 by reviewing the tasks identified in Chapter 4 and identify methods to measure the result quality of these tasks in Section 7.1. We answer Question 5 by conducting an in-lab user study on some of the identified tasks using real world data, reported in Sections 7.2 and 7.3. We include a description of how we acquired the data and also explain limitations of our approach. We then present the results of the study. This chapter concludes with analyzing the results of the study in Section 7.5 and a conclusion detailing the answer for Research Question 5.

7.1 Quality Measures

In Chapter 4, we identified the following tasks as being essential for knowledge workers using digital content: (Co)-Authoring Information, Information Search, Information Organization, Information Dissemination and Versioning.

7.1.1 Measurements

Existing quality measures (term frequency and inverse document frequency) from the field of Information Retrieval focus on establishing how much a document reflects specific content, such as a string of words. However, these quality measures are only applicable when searching for specific content, i.e. are part of task of Information Search. Here we are instead concerned with the result quality of our tasks. We suggest to measure them by the time needed to complete them, the accuracy of the outcome and by how much confidence the conductor of the task has in the end result of the task.

Speed The speed at which a task can be completed is an indicator for the overall result quality of the task. The accuracy of the result of a task matters little if the time taken to complete the task was to long and the result was determined to slow. Measuring the speed at which the identified tasks are completed is straightforward since there are examples for these tasks that can be completed in a short time span, with the possible exception of (Co)-Authoring Information. If a task can be executed quickly but with low accuracy, its result quality is still considered low.

Accuracy The accuracy of the outcome of a task is another indicator for task result quality. If a task can be executed quickly but with low accuracy, its result quality is still considered low. The feasibility of measuring the accuracy of the identified tasks varies. A naive measure of the accuracy of finding information is either success or failure in finding the searched information. Organizing information, disseminating information and versioning information are used to keep information accessible to knowledge workers. Therefore the accuracy for these tasks can be measured in the same way that we measure accuracy for information search. Measuring the accuracy of (Co)-Authoring Information is difficult, as the outcome of

that task can be classified as more than binary success or failure. However, we believe that the other tasks are often used to achieve the goal of (Co)-Authoring Information. Therefore, by measuring the accuracy of these tasks, we gain an indication of the accuracy of (Co)-Authoring Information.

Confidence The confidence of the task conductor in the final result has an effect on the overall result quality of the conducted task. If the task conductor does not feel confident in the outcome, they may feel compelled to redo the task, therefore increasing the time needed to complete the task. Measuring the confidence of a user can be done through the use of Likert scales. However, we need to be aware that such results are not precise, as users' definitions of high and low confidence can differ.

7.1.2 Validity of Results — In-Lab Setting

When measuring the result quality of tasks the validity of the results depends on the authenticity of the tasks. Observing knowledge workers whilst they perform their tasks at work is not feasible for this work for two reasons. Firstly, the act of observing a knowledge worker either influences how the knowledge worker performs tasks when triggered to perform them, or because it takes a large amount of time for the observer to wait and observe the task when it occurs 'naturally'. Secondly, the prototype software was not allowed to be installed to life work environments. We contacted several companies interested in our research approach, but none was able to allow us to install our prototype on the knowledge workers' devices.

We therefore conducted the observation of the tasks in an in-lab setting. To reach the best possible authenticity of the observed tasks, we needed to make sure that the observed tasks and the data used are taken from real scenarios.

7.2 Data Acquisition

In order to gain authentic data, we needed a knowledge worker who uses Microsoft Word to maintain and create documents, so we could install and run the prototype on that knowledge worker's PC. The university allowed us to install our prototype on the PC of knowledge workers under the

condition that they and their line managers agreed.

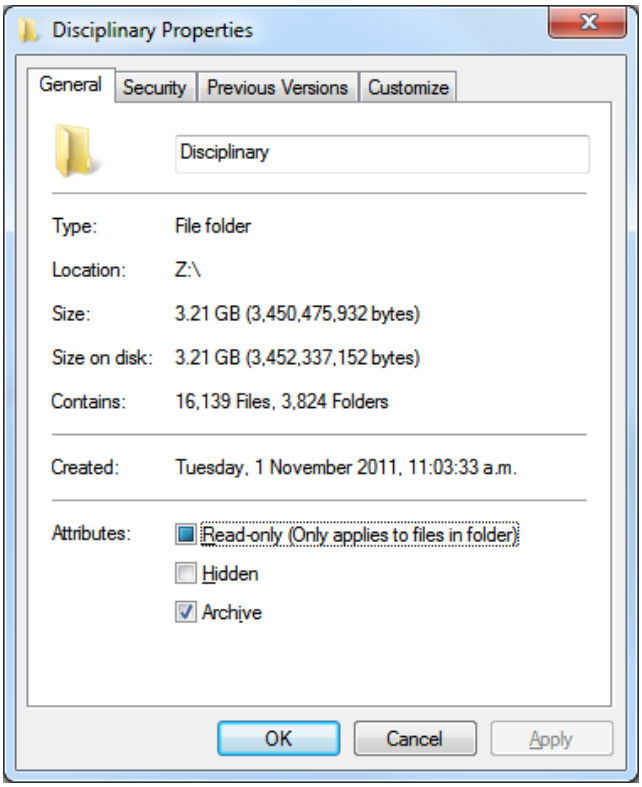


Figure 7.1: File Count of the Knowledge Worker

We contacted a knowledge worker willing to participate in our research and installed the DDNA Tracker on their work PC in December 2012. We met the knowledge worker again in June 2014 and installed the DDNA Analyzer. We asked the knowledge worker to show us a few examples of tasks that they had performed since we installed the DDNA Analyzer.

One of the knowledge worker's roles was to support the Students Disciplinary Committee and to maintain associated documents. Figure 7.1 shows a file count on the folder used to store all related files. The total number of files contained is 16,139 in 3,824 folders.

7.2.1 Tasks Performed

The participating knowledge worker named the following tasks when describing their work: Analyze Information, Acquisition of Information, Disseminate Information and Information Organization. In regards to the supporting role for the committee, information was mostly requested and disseminated via letters, memos or handouts.

Therefore, the knowledge worker maintained a large number of templates for different document types they were using. These templates often included references to the university's calendar and guidelines, and therefore required updating whenever the source of the reference is updated. These updates were usually processed in waves once a year. More commonly, the knowledge worker created and mailed letters to distribute information relating disciplinary cases or appointments. These letters would then be saved in digital form for future references.

7.2.2 Data Selection

We chose two self-contained scenarios from the scenarios the knowledge worker showed us.

Scenario 1 — Disciplinary Case

The first scenario includes 22 result letters for a disciplinary case processed between 2013 and 2014. Each student involved in this case received one result letter. The letters have been anonymized by replacing all dates and names with consistent place holders. This scenario also includes all 122 templates used to create documentation of disciplinary cases.

Figure 7.2 shows all relationships between the letters and templates. All letters were created with the use of the template `Result letter misconduct in a test SDC` and all but two letters use content from the template `Result letter plagiarism SDC`. There are many missing document vertices, since the letters were edited often and we did not have access to the archive of these versions. Figure 7.3 shows the relations of the documents without the missing versions in between. Note that both graphs have been manually arranged to fit the figure.

The templates are stored in the folder named `Discipline letter templates` and the letters are stored in the folder `January 2014`. We decided to condense the surrounding folder structure to these two folders, due to the high number of documents involved in the scenario.

Scenario 2 — Summary Jurisdiction Appointments

The second scenario includes the correspondence for appointing staff members to summary jurisdiction duty. These appointments last one year and

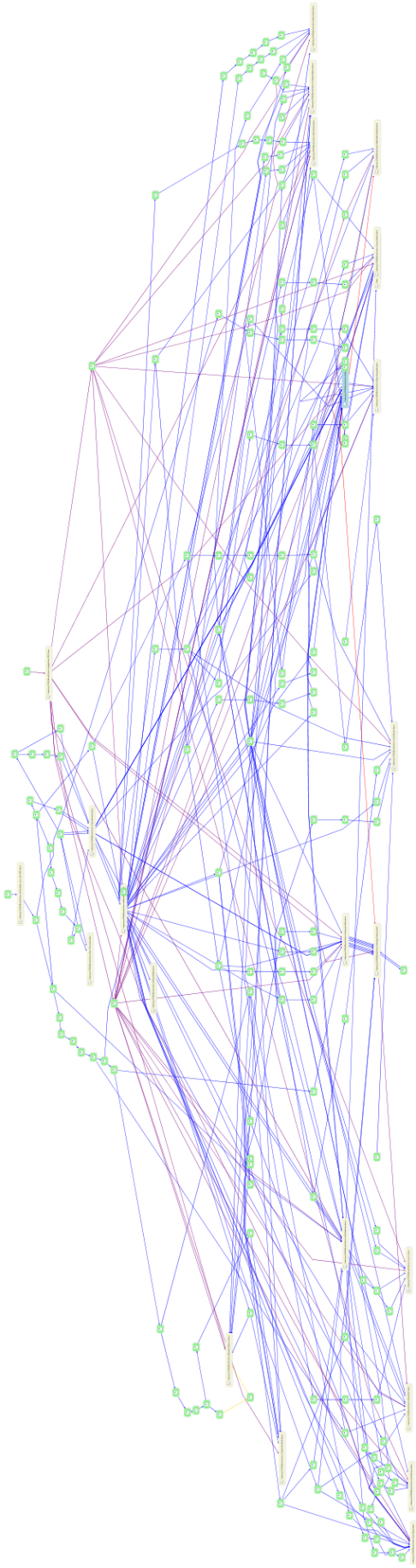


Figure 7.2: Document Relations for Scenario 1

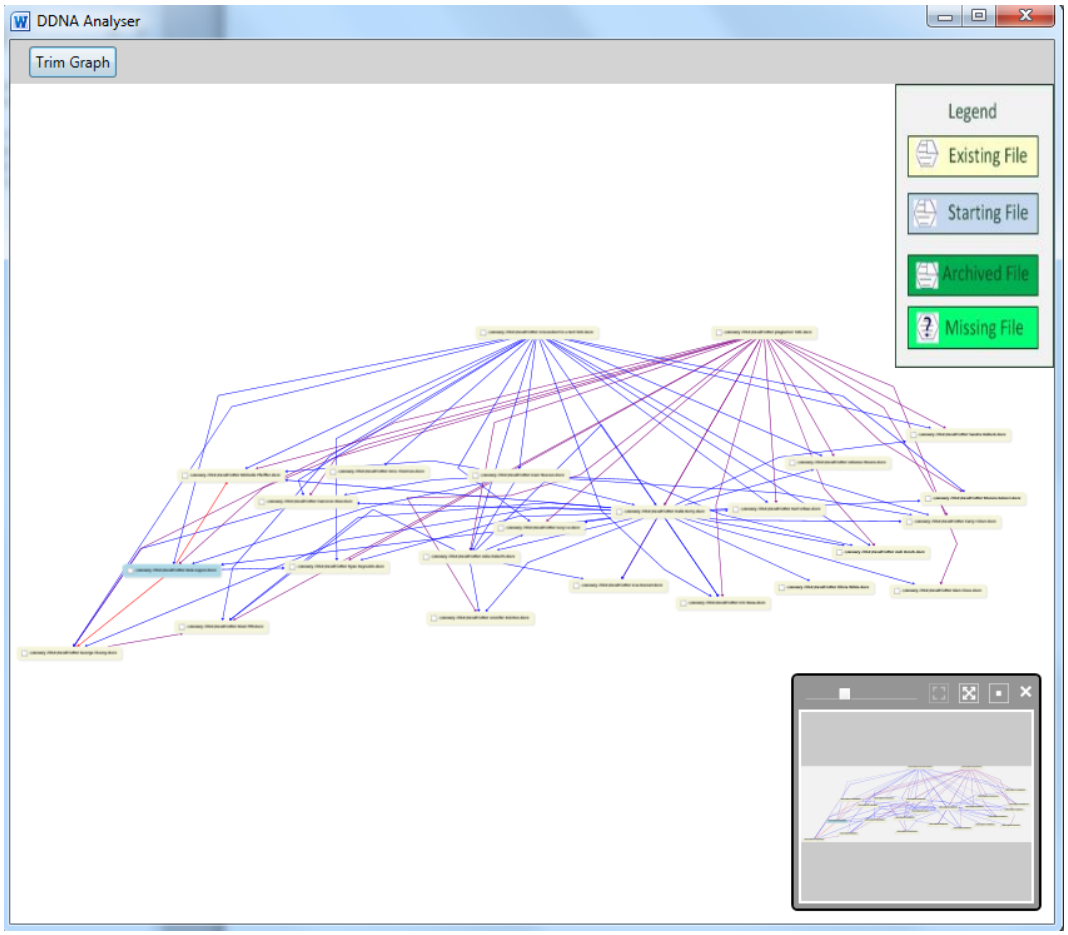


Figure 7.3: Document Relations for Scenario 1 — Trimmed

each appointment is affirmed with a confirmation letter. Staff members appointed to the position receive formal training. Once such training has been received they do not need to train again if they return to the position in following years. Confirmation letters for new appointments therefore differ slightly from confirmation letters for re-appointments.

Sent letters are kept in sub-folders of the folder `Confirmation Letters`. These sub-folders are named by years, such as `2013`. The main folder also includes two templates, `New appointment confirmed` and `Re-appointment confirmed`, to create the letters.

We only kept the letters for this scenario that were created when the DDNA Tracker was active, therefore limiting the range to 2013–2014. Figure 7.4 shows relationships between the documents contained in the folder `Confirmation Letters`. Note that the graph has been manually arranged to fit the figure. There are nine letters for new appointments, three

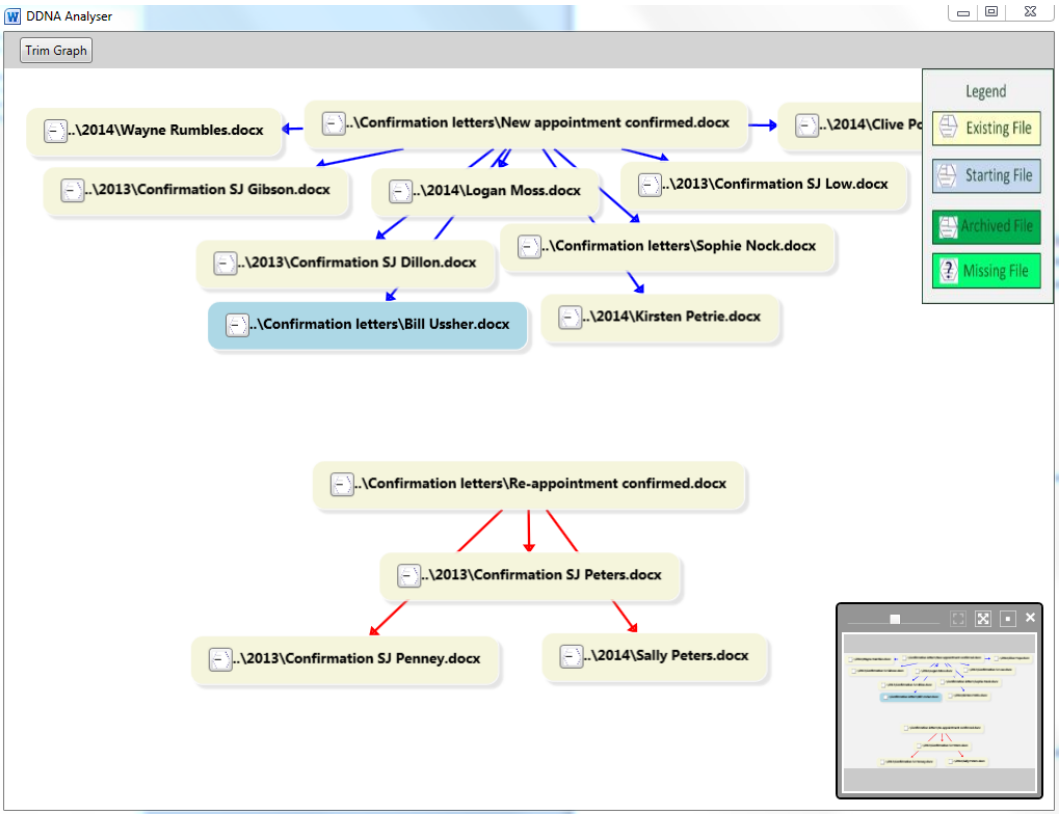


Figure 7.4: Document Relations for Scenario 2

letters for re-appointments and two templates. Two of the letters for new appointments have not yet been sorted into the appropriate folder.

For more realism, we also included the top folder of the folder `Confirmation Letters` which includes seven folders and 81 files, including the ones from the folder `Confirmation Letters`. The folder includes more folders and files of administrative nature, but no other data concerning staff appointed for summary jurisdiction.

7.3 Study Design

This study was composed of two tasks that two groups of participants were asked to execute. The first group of participants were a control group and were asked to execute the tasks on a standard PC with Microsoft Windows 7 and Microsoft Office 2013 installed, as shown in Figure 7.5. The second group of participants were asked to execute the same tasks, but were introduced to the DDNA Analyzer beforehand and were encouraged to use the DDNA Analyzer. Additionally to the two tasks, we asked each participant

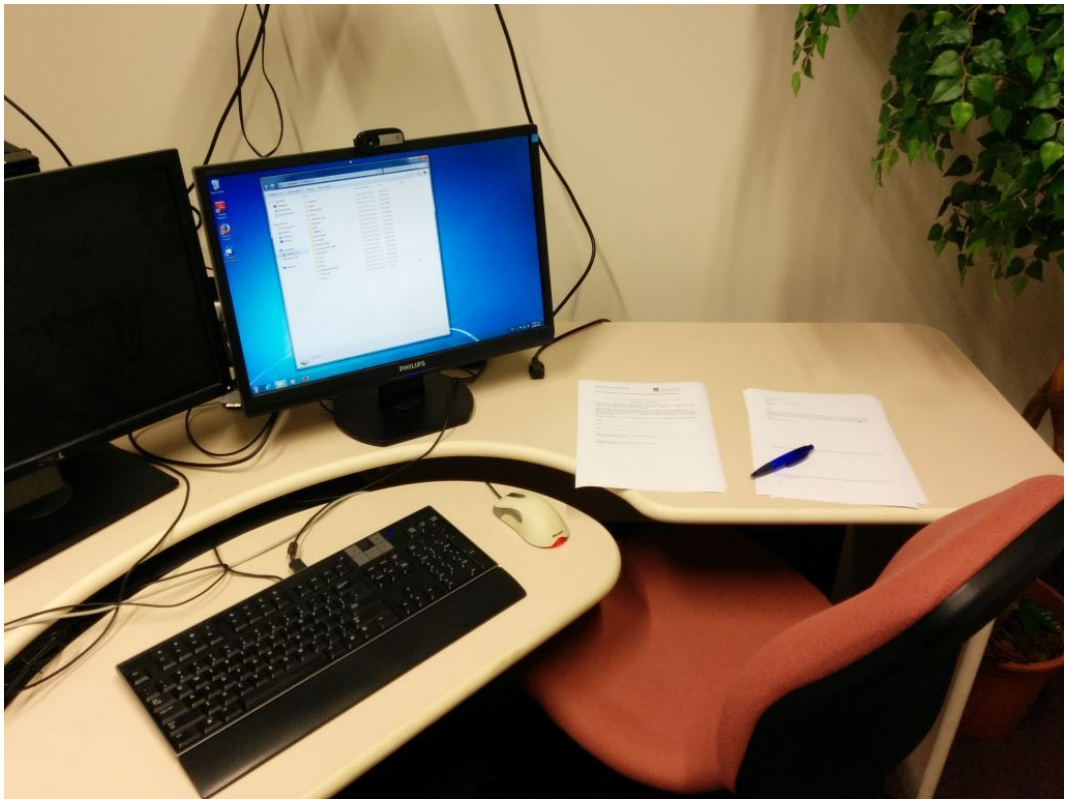


Figure 7.5: Workplace Used for the Study

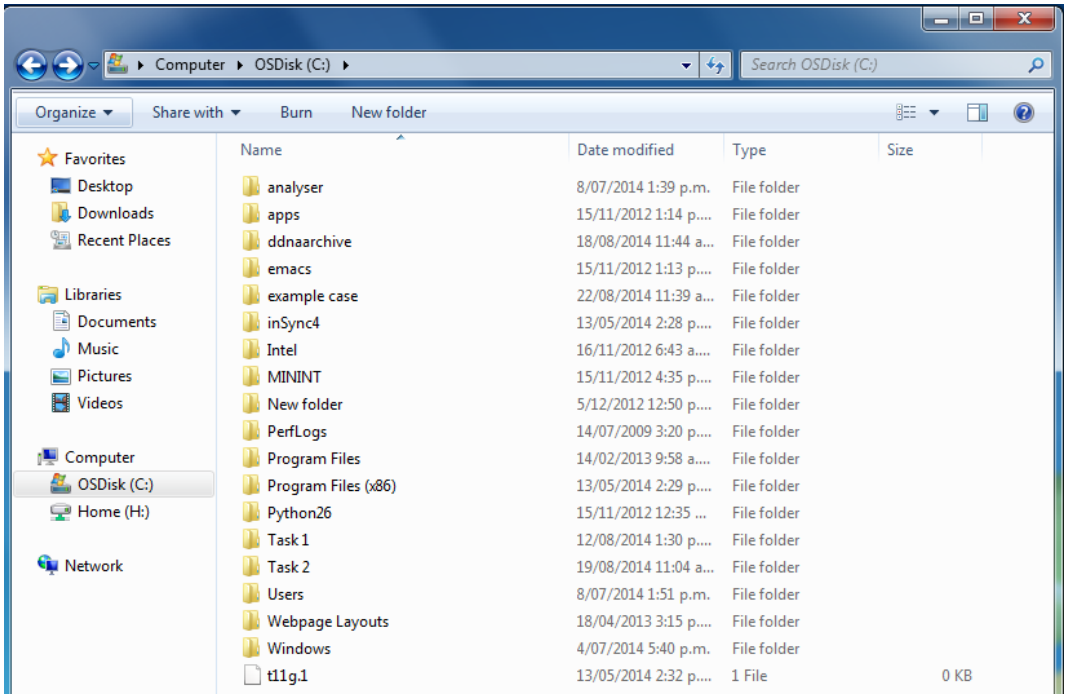


Figure 7.6: Explorer Setup

for their age, gender and profession.

Participants We approached students and staff at the university to gain participants for this study. We had no requirements for participants except being fluent in English and being familiar with a Windows 7 PC.

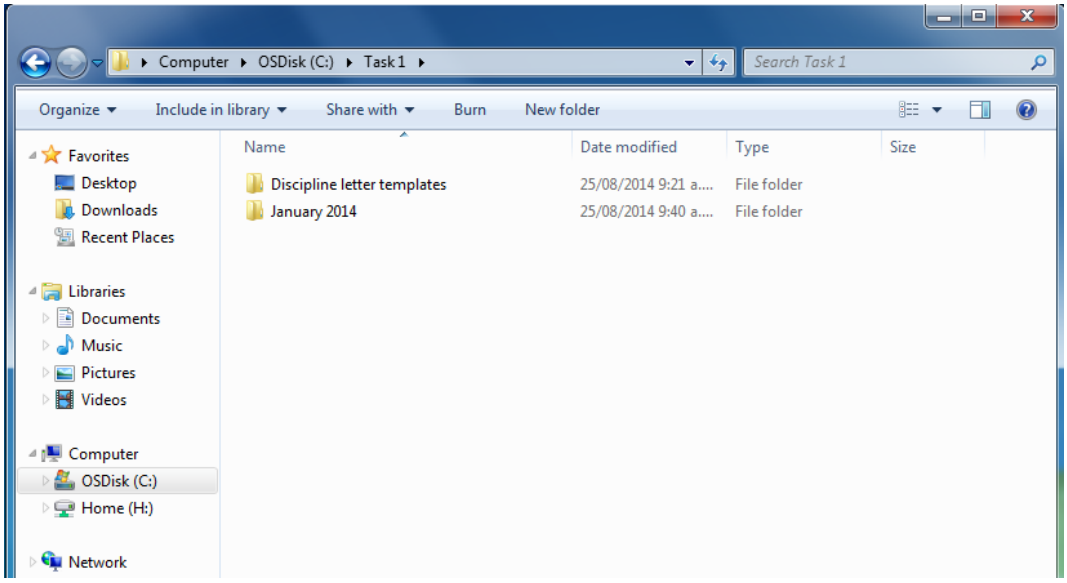


Figure 7.7: Explorer Setup — Task 1

Lab Setup This study utilized the Usability Lab of the Department of Computer Science of the University of Waikato. Figure 7.5 shows the general set up of the work place, which is designed to simulate an office environment. The PC supplied had Microsoft Windows 7 and Office 2013 installed. We also installed the DDNA Analyzer on the PC. No other software was installed.

The scenario data for each task was provided in a folder labeled `Task1` or `Task2` on the C drive, as seen in Figure 7.6. The folders contained the structure from the earlier described scenarios, as shown in Figures 7.7 and 7.8.

7.3.1 Tasks

We asked the participants to execute two tasks. The aim of these tasks was to simulate a scenario where a knowledge worker needs to continue the work of another knowledge worker. We finished both tasks by asking the participant how confident they were that the outcome of their task was complete and correct.

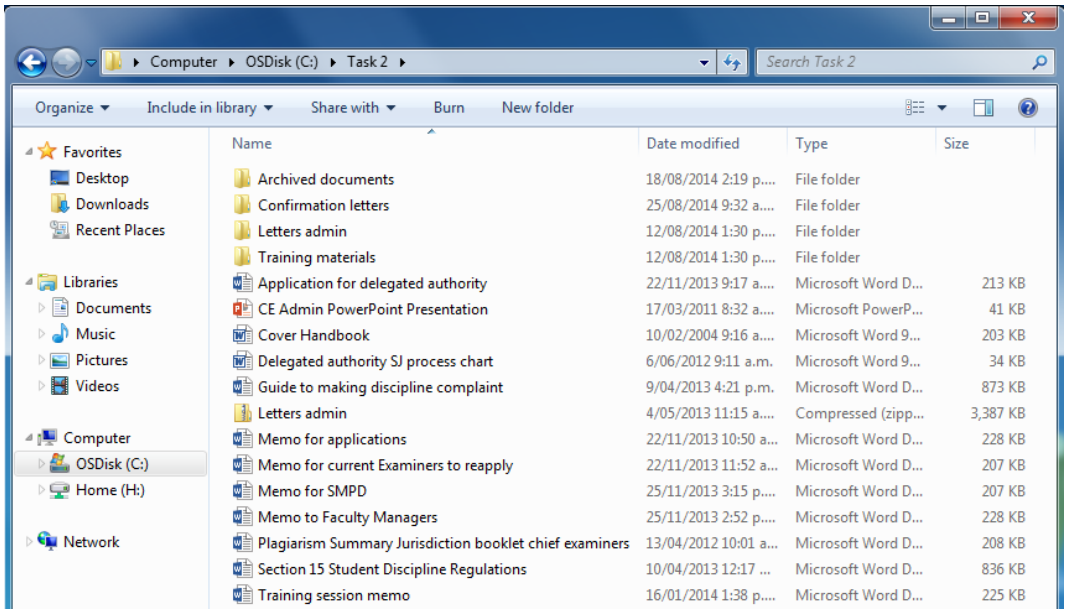


Figure 7.8: Explorer Setup — Task 2

Task 1 The first task had two parts:

1. The document `C:\Task1\January 2014\Result letter Bela Lugosi.docx` was created with the help of template(s) from the folder `C:\Task1\Discipline letter templates`. This means that content was copied and pasted from the templates to the letters, or the template edited into a letter and saved as the letter. Which templates were used?

2. What other files in the folder `C:\Task1\January 2014\` were created with the use of these templates?

The aim for the first part of the task was to observe if participants were able to identify which templates are needed to create letters for a similar disciplinary case. The aim for the second task was to observe if participants were able to identify which other letters were created using the found templates.

For the control group, the naming conventions of the letters and templates were not sufficient to solve the task. The content of the letters also needed to be compared with the template and the correct key words: 'Result', 'SDC', 'misconduct', 'plagiarism', and 'quiz' needed to be detected. Participants were given 12 minutes to complete the task. We intentionally

chose a narrow time limit to simulate time scarcity at the work place.

Solution The correct templates were `Result letter misconduct in a test SDC` and `Result letter plagiarism SDC`. All letters in the letter folder contained content from at least one of the templates.

Task 2 The second task had the following question:

1. The folder `C:\Task2\` includes all confirmation letters and templates for staff appointed to undertake Summary Jurisdiction hearings. Please list staff that was re-appointed and staff that was newly appointed for the role and when.

The aim of the task was to observe if participants were able to identify two different groups of letters and the template used to create each letter. The naming conventions for both letters and folders included no information regarding this, the only way to detect the difference was by comparing the content. The second aim was to observe if participants were able to identify the time the letters were sent and which person was the recipient of the letter.

For the control group, this information was partially available in the naming of the files and folders, as files always included the last name of the recipient and folders included the year the letter was sent. However, not all files were sorted into the appropriate folder. Participants were given eight minutes to complete the task. Again, the narrow time limit was chosen intentionally to simulate time scarcity.

Solution Six staff were newly appointed in 2014 and three in 2013. Two staff were re-appointed in 2013 and one staff member was re-appointed in 2014. The documents `New appointment confirmed` and `Re-appointment confirmed` were the templates used to create the letters.

7.3.2 Training Scenario

To introduce the participants to the DDNA Analyzer, we demonstrated both functions of the DDNA Analyzer on a pre-created example set of documents. The example scenario consisted of six documents in one folder: `A`, `A2`, `B`, `C`, `D` and `Merger`. The relationships of the documents are shown in Figure 7.9. We did not include any archived documents in the example

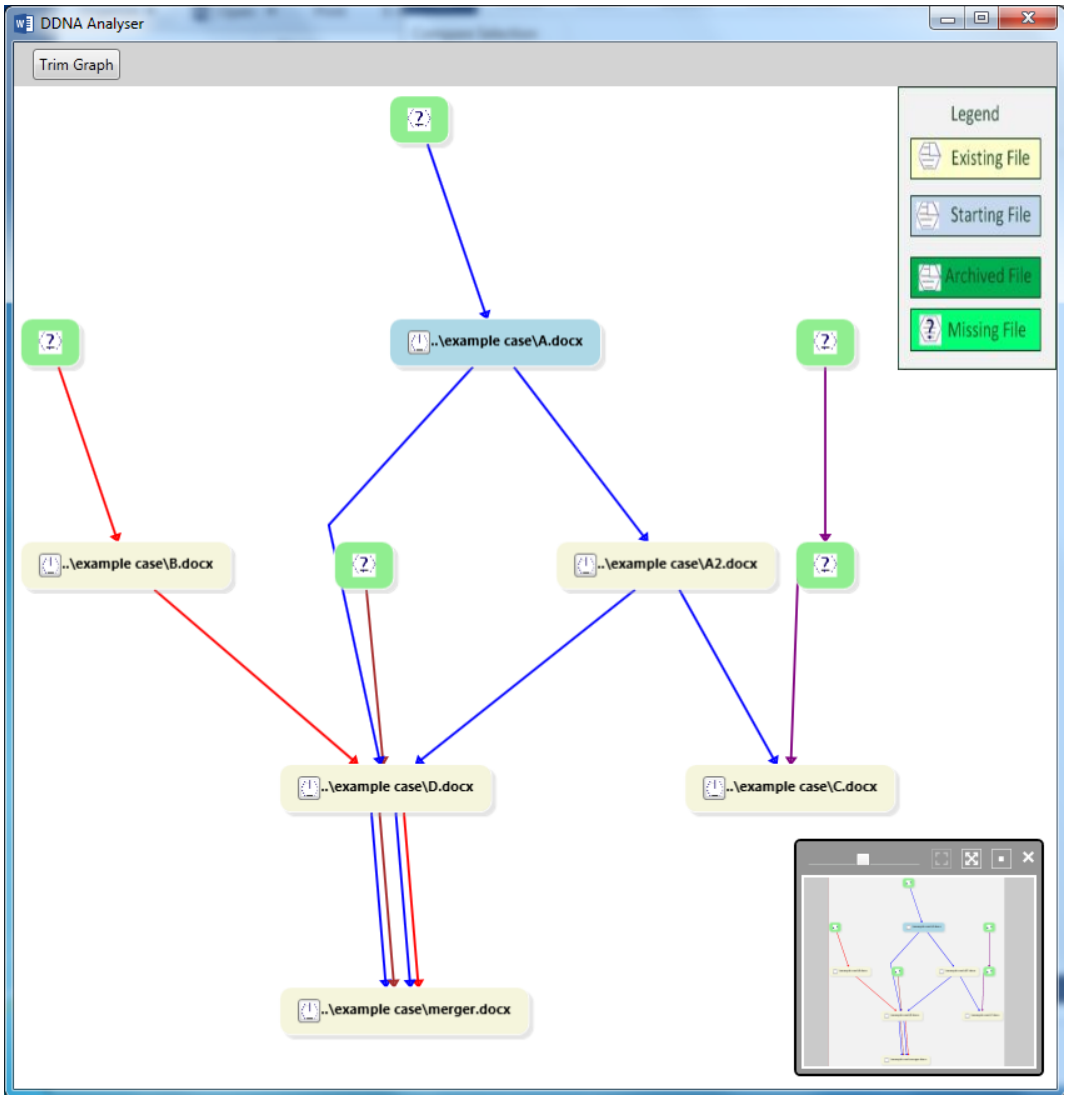


Figure 7.9: Relationships of Training Scenario

scenario, as no archived documents were used for the user study tasks.

7.4 Results

We now present the results of this study, starting with the participants' statistics. We follow with the result of the control group and finish with the group using our software prototype.

7.4.1 Participants

To ensure anonymity, participants were given the identifiers P1–P30. P1–P15 were in the control group and P16–P30 were in the DDNA group. This

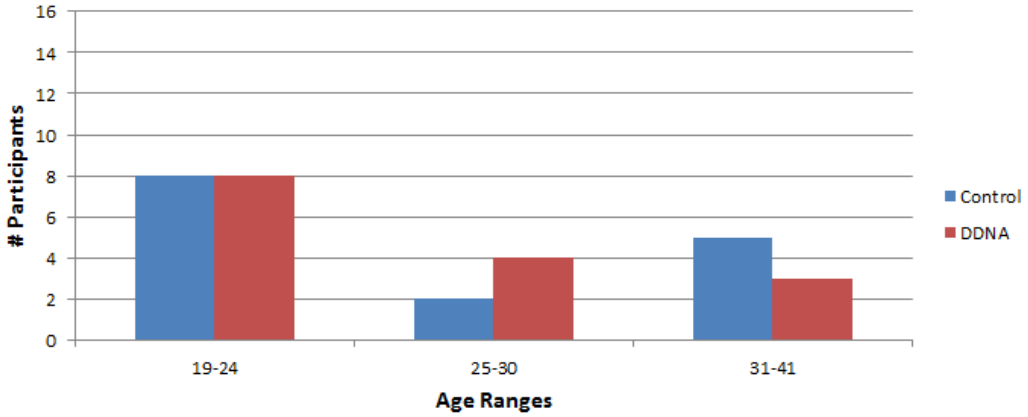


Figure 7.10: Participants' Age Ranges

study had 30 participants overall, 26 students, two lecturers (P19 and P23), one research fellow (P16) and one research coordinator (P4).

The study had seven female and 23 male participants. The average age of the control group was 25.8 years and the average age of the non-control group was 24.8 years. The age ranges for both groups are shown in Figure 7.10. We did not detect any gender or age relation for our results.

7.4.2 Task Results — Control

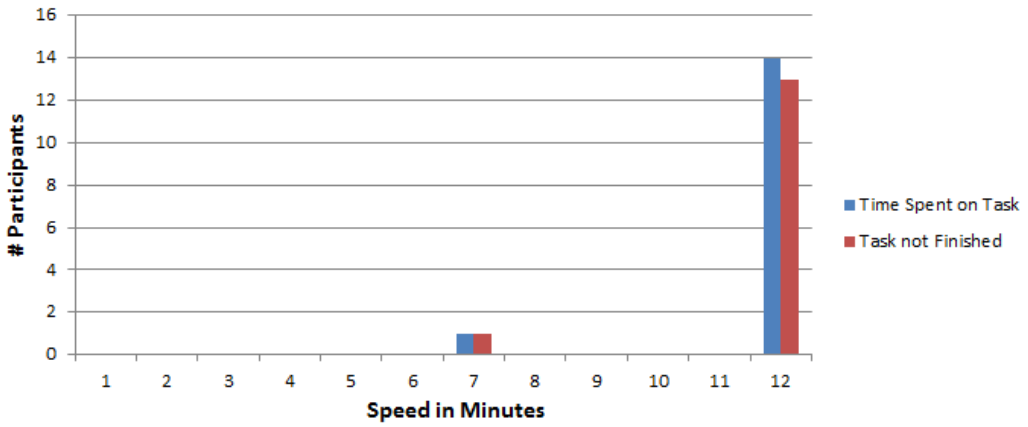


Figure 7.11: Time Spent on Task 1 — Control Group

The control group was asked to complete Task 1 and 2 on a standard Windows 7 PC.

Task 1

Speed The times taken to complete Task 1 are shown in Figure 7.11. All participants but one (P8) used the full time limit for the first task. However, no participant fully completed the task in the given time except P4. P8 did not use the full time limit as they did not know how to proceed any further and stopped after contemplating ways to solve the task after 7 minutes.

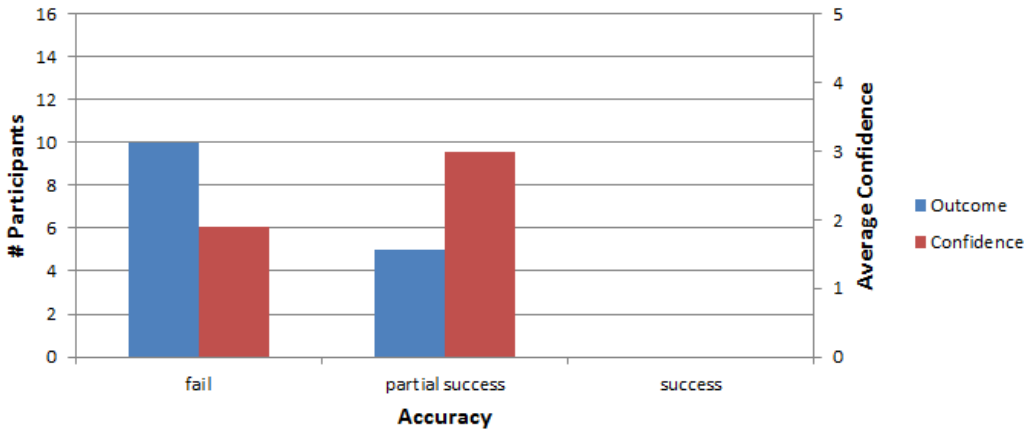


Figure 7.12: Accuracy of Task 1 — Control Group

Accuracy The achieved outcomes for Task 1 are shown in Figure 7.12. No participant solved the task completely and only five participants (P6, P9, P11, P12 and P13) achieved partial success, listing one of the involved templates. No participant solved the second part of Task 1.

Confidence Confidence was measured from 1 (no confidence) to 5 (fully confident). Figure 7.12 shows the average confidence participants expressed after executing the task. The overall average was 2.3. The participants failing the task expressed an average confidence of 1.9 and the participants partially succeeding expressed an average confidence of 3.

Task 2

Speed The times taken to complete Task 2 is shown in Figure 7.13. Two participants (P6 and P14) completed the task within 6 minutes and two participants (P2 and P10) completed the task in 7 minutes. All other participants spent 8 minutes on the task, but only five of those participants (P1, P3, P4, P9 and P12) completed the task.

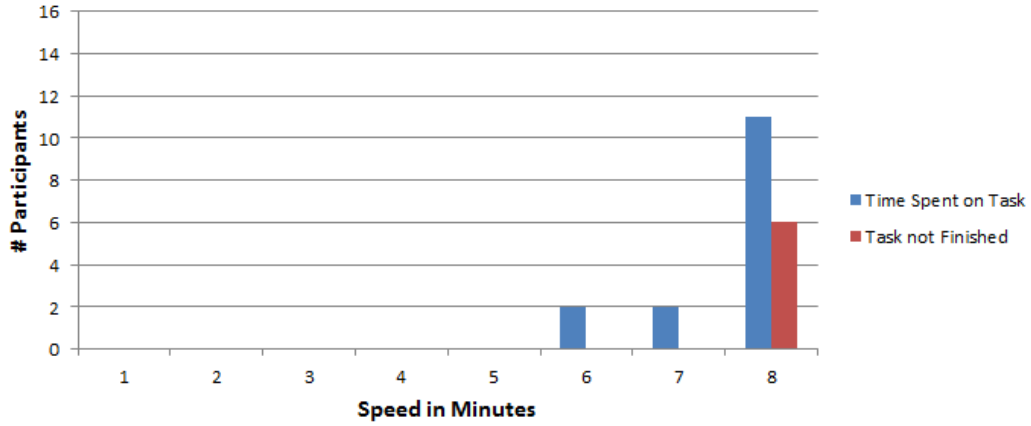


Figure 7.13: Time Spent on Task 2 — Control Group

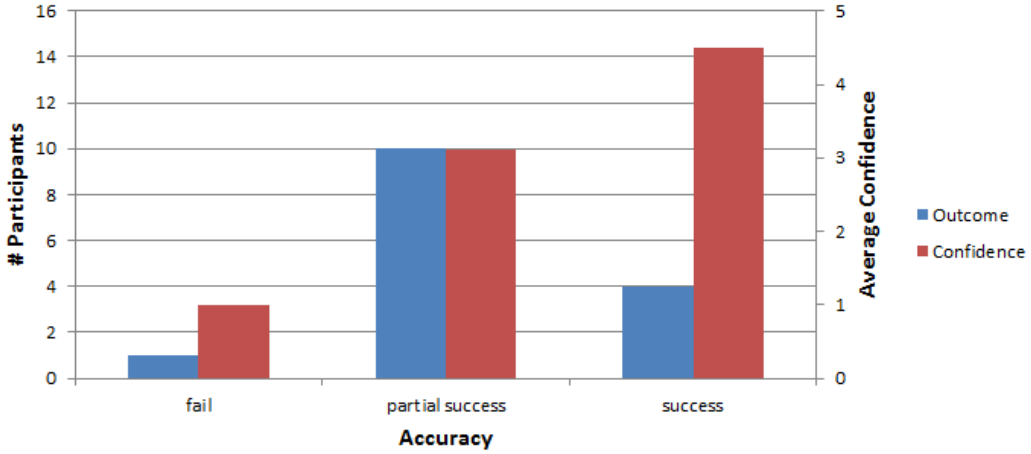


Figure 7.14: Accuracy of Task 2 — Control Group

Accuracy The outcomes for Task 2 are shown in Figure 7.14. Only one participant failed the task (P11) and four participants (P2, P7, P10 and P14) succeeded. The rest of the participants achieved partial success by identifying all the names of staff appointed to Summary Jurisdiction, but failed to either name whether or not they were newly appointed, the year of the appointment or both.

Confidence Figure 7.14 shows the average confidence participants expressed after executing the task. Confidence was measured from 1 (no confidence) to 5 (full confidence). The overall average of confidence expressed for Task 2 was 3.33. The average for confidence for participants who achieved success was 4.5 and the average for participants who achieved partial success was 3.1. The participant who failed the task (P11) expressed

a confidence of 1.

Opened Documents

As another measure of result quality, we observed how many documents were accessed by the study's participants.

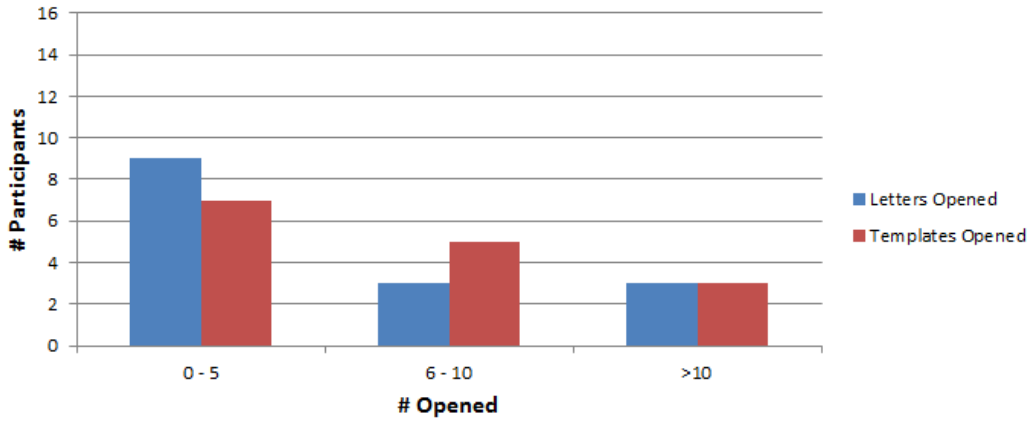


Figure 7.15: Documents opened for Task 1 — Control Group

Opened Documents — Task 1 The maximum amount of documents accessible for Task 1 were 22 letters in the folder `January 2014` and 122 templates in the folder `Discipline letter templates`. The participants opened at least 1 and at most 22 Letters. The minimum opened for templates was 0 and the maximum was 122.

Figure 7.15 shows that only three participants opened more than 10 letters (P1, P3 and P13) and also only three participants opened more than 10 templates (P1, P3 and P11). Three participants opened 6-10 Letters (P2, P5 and P15) and five participants opened 6-10 templates (P2, P6, P11, P14 and P15). The other participants opened in between 0 and 5 letters and between 0 and 5 templates.

Opened Documents — Task 2 The maximum amount of letters accessible were 14 Letters (2 of them templates) in the folder `Confirmation Letter` and 67 other documents spread across 4 folders. The Participants opened a minimum of 2 letters and a maximum of 14 letters. Access to the other unrelated files was not counted, as only three participants (P4, P13 and P15) browsed files outside of the folder `Confirmation Letter` and opened between 1–3 unrelated documents.

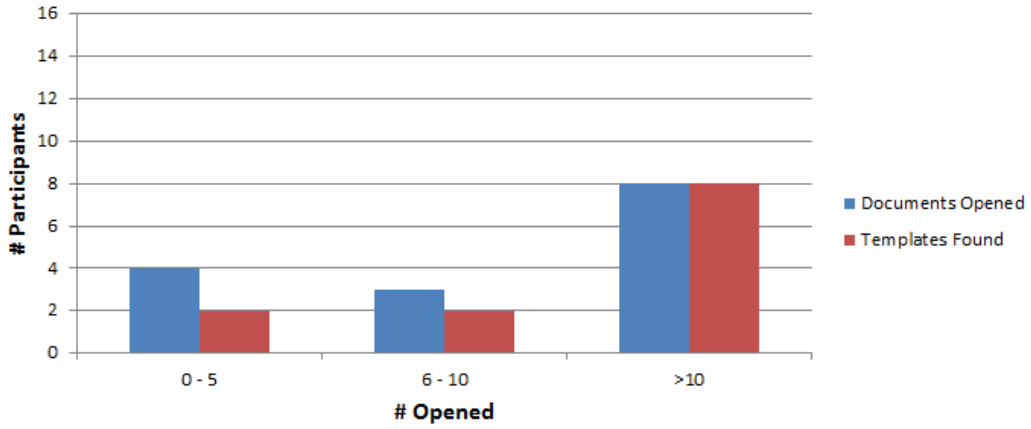


Figure 7.16: Documents opened for Task 2 — Control Group

Figure 7.16 shows how many letters the participants opened and if those participants identified the templates. Four participants opened 0-5 letters (P4, P5, P8 and P13), but only two of those participants identified the templates (P8 and P13). Another three participants opened 6–10 letters (P7, P9 and P11) and only P11 did not identify the templates. P15 opened 12 letters and identified the templates. The rest of the participants opened all 14 letters and identified the templates.

Opened Documents — Success? We could not identify a correlation between finding the right template and the number of opened templates and letters for Task 1. P13 needed only to open three templates but opened 22 letters to find one correct template, whilst P10 opened 30 templates and one letter to find one correct template. However, all four participants that succeeded Task 2 opened all 14 letters and identified the correct templates.

Participants' Strategies

We observed the participants whilst conducting Task 1 and 2 and found that the participants used different strategies on how to approach the task. We structured the strategies we observed by what participants relied on: Content Structure, File-Folder Structure, Content, Explorer Search and File Properties, and Word Functionality. Figure 7.17 shows how many participants used the different strategies.

Content Structure The content structure strategy was used by twelve participants, only P1, P5, and P8 did not use it. Participants using this strategy compared the structure of the content in the letters to the structure of the

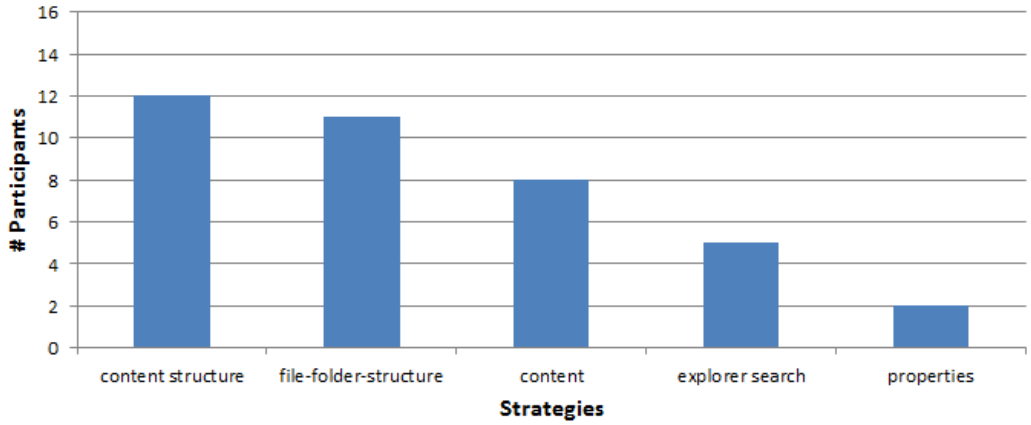


Figure 7.17: Strategies — Control Group

content in the templates in order to determine whether or not a template was used. The structure of content refers to the content of headings, the placement of headings, points included in bullet lists and the general size of paragraphs.

File-Folder Structure This strategy was used by eleven participants, only P1, P5, P7, and P8 did not use it. Participants using this strategy skimmed the name and the contents of the starting letter for keywords like 'Outcome', 'Quiz' or 'Misconduct'. These keywords were then searched for in the titles of the templates, including synonyms of the keywords. Only P6, P13, and P14 realized that the abbreviation 'SDC' in file names meant 'Student's Disciplinary Committee', which was an important indicator towards the right templates.

Content Eight participants (P4, P6, P7, P11, P12, P13, P14, and P15) used the content strategy. This strategy means that the participants compared the content from the letter and the templates, such as wording of similar paragraphs or numerical values.

Explorer Search Five participants (P1, P5, P7, P10, and P11) used the search function build into the Windows Explorer. This strategy included searching for keywords like 'Outcome', 'Quiz', or 'Misconduct', but also included search for full sentences from the starting letter.

File Properties & Word Functionality Participants P7 and P15 used this strategy. It included comparing meta data accessible through the Windows Explorer like page count, file size, or file type to spot similarities. This

strategy also included utilizing Microsoft Word features to compare documents or browse the version history of documents. The participants did not know that the documents were created without the active use of such features, which greatly limited the use of these features.

Winning Strategy No single strategy could be identified that would always lead to success. However, our observation was that participants made most of their progress towards finishing the task by using the second and third strategy, whilst the 4th and 5th strategy yielded no progress at all.

7.4.3 Task Results — DDNA

The non-control group was asked to complete Task 1 and 2 on a standard Windows 7 PC on which the DDNA Analyzer was installed. We introduced the participants to the DDNA Analyzer via the training scenario and started the study after they had no more questions.

Task 1

Speed All but one participant (P30) were able to complete the task within the given time. Figure 7.18 shows that P22 needed three minutes, two participants (P26 and P28) needed four minutes, P24 needed five minutes, two participants (P17 and P26) needed six minutes, two participants (P19 and P27) needed seven minutes, P23 needed eight minutes, four participants (P16, P20, P21, and P29) needed nine minutes and two participants (P18 and P30) needed the full twelve minutes.

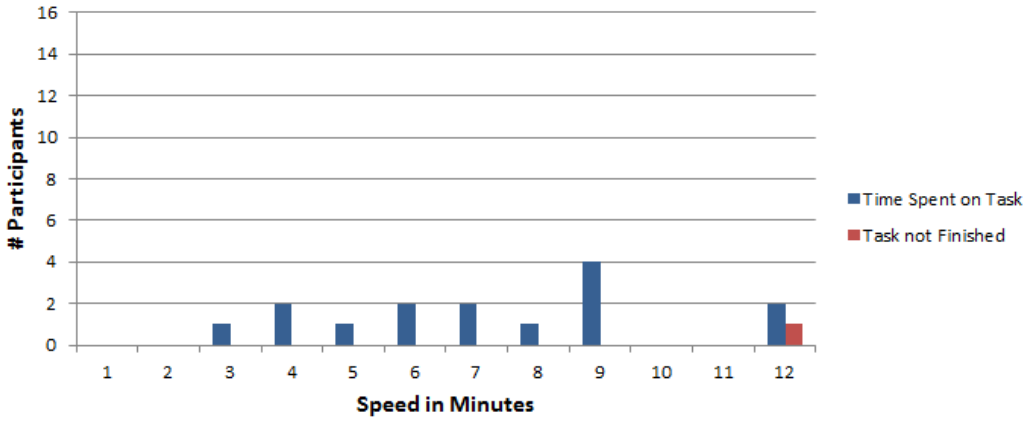


Figure 7.18: Time Spent on Task 1

Accuracy The outcomes for Task 1 are shown in Figure 7.19. Two participants (P17 and P30) achieved partial success, as they were not able to name all letters created by the found templates. All other participants found both templates and determined that all letters in the letter folder were created using at least one of the templates.

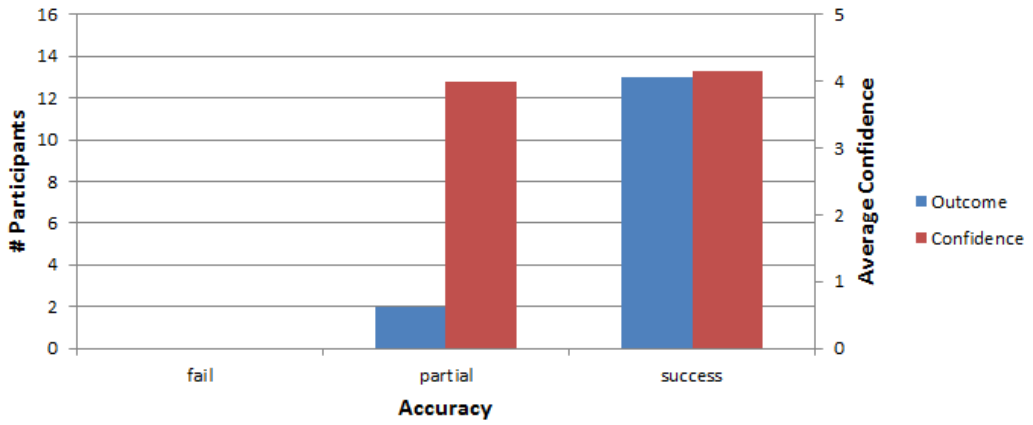


Figure 7.19: Accuracy of Task 1

Confidence Confidence was measured from 1 (no confidence) to 5 (full confidence). Figure 7.19 shows the average confidence participants expressed after executing the task. The overall average was 3.8. The participants who partially succeeded expressed an average confidence of 4 and the participants who succeeded expressed an average confidence of 3.8.

Task 2

Speed All participants were able to complete the task within the given time. Figure 7.18 shows that five participants (P17, P18, P19, P25, and P28) needed five minutes, three participants (P22, P27, and P29) needed six minutes, four participants (P16, P21, P26, and P30) needed seven minutes and three participants (P19, P23, and P24) needed eight minutes.

Accuracy The outcomes for Task 2 are shown in Figure 7.21. Two participants (P18 and P21) achieved partial success, as they failed to identify the correct years for some letters. All other participants correctly identified all appointed staff, whether or not the appointment was new, and which year the appointment took place in.

Confidence Confidence was measured from 1 (no confidence) to 5 (fully confident). Figure 7.21 shows the average confidence participants expressed

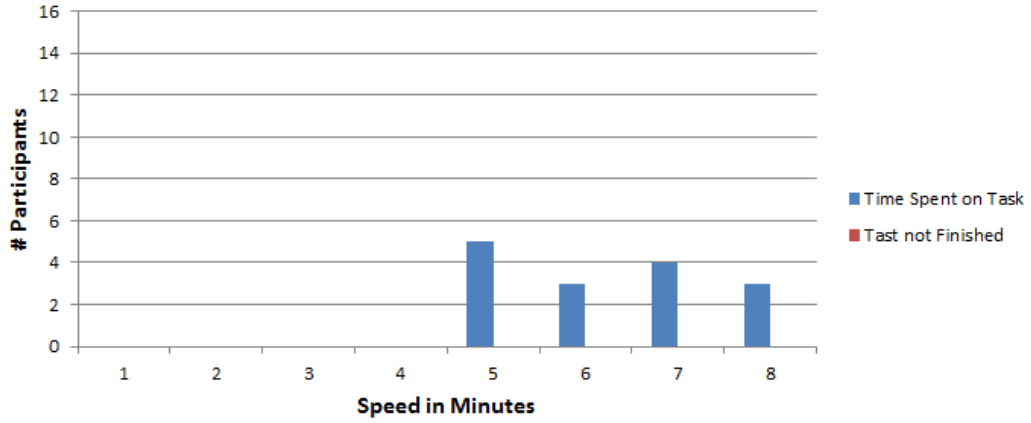


Figure 7.20: Time Spent on Task 2

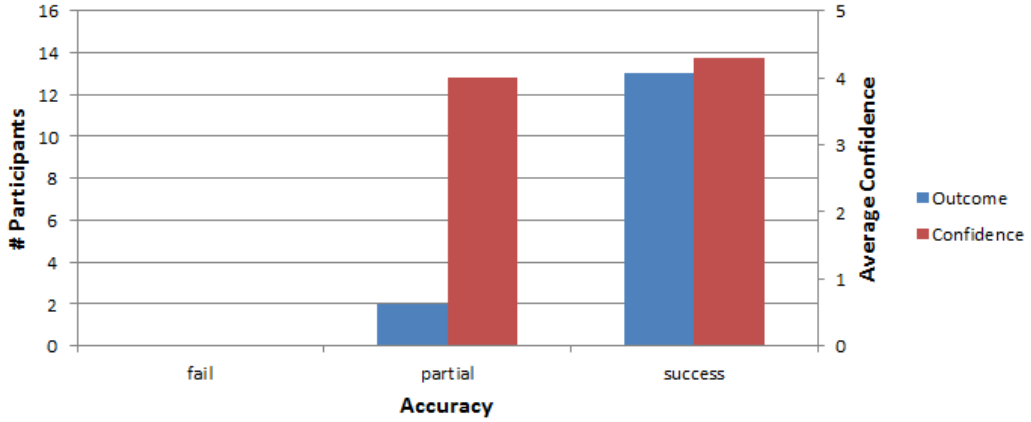


Figure 7.21: Accuracy of Task 2

after executing Task 2. The overall average was 4.1. The two participants who partially succeeded expressed an average confidence of 4 and the 13 participants who succeeded expressed an average confidence of 4.2.

Opened Documents

Opened Documents — Task 1 All participants opened less than five letters and templates, as shown in Figure 7.22. All participants but P30 opened one letter, P30 opened two. Four participants (P17, P18, P20 and P27) opened no template and two participants (P16 and P23) opened two templates. All other participants opened one template.

Opened Documents — Task 2 Figure 7.16 shows how many letters the participants opened and if those participants identified the templates. All participants identified the templates and all but one participant (P28) opened

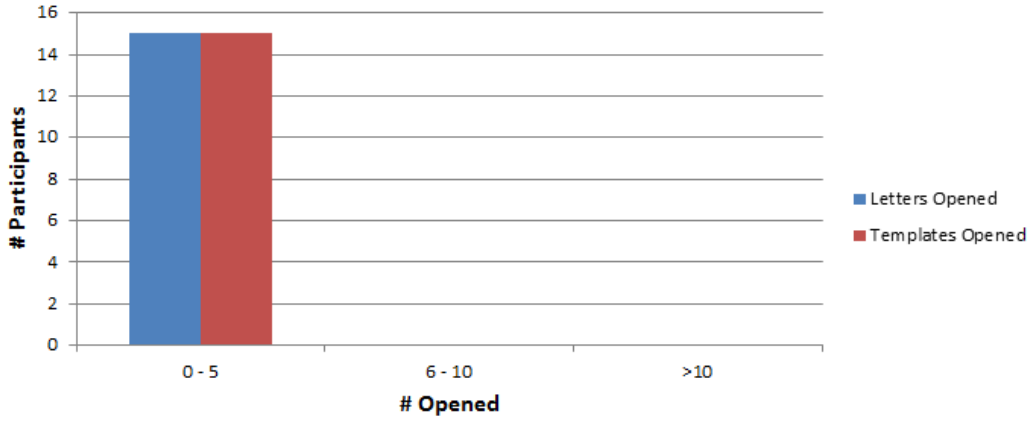


Figure 7.22: Documents opened for Task 1

one to five documents. The participants opened a minimum of three documents (P18, P21, and P27) and a maximum of ten documents (P28). Two participants (P22 and P30) opened five documents and all other participants opened four documents. No participant opened documents that were not confirmation letters or templates.

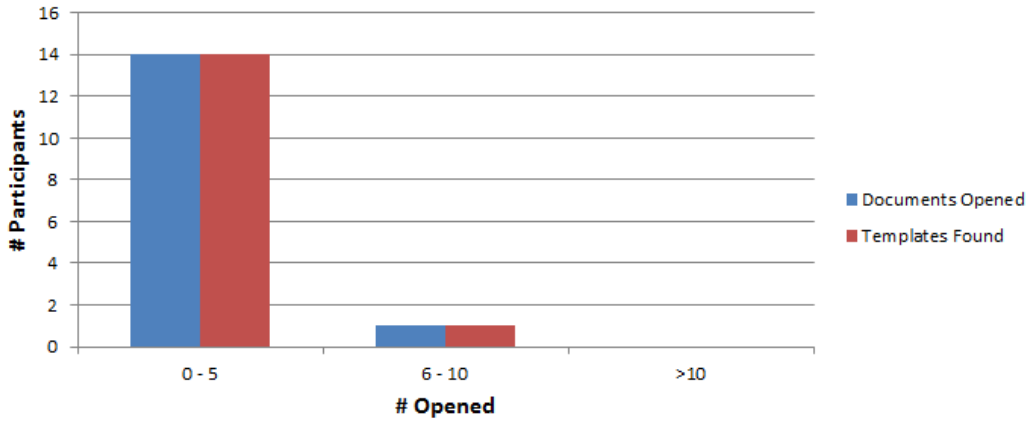


Figure 7.23: Documents opened for Task 2

Opened Documents — Success? We observed participants succeeding that opened both the minimum and the maximum observed for documents for Task 1 and Task 2. We therefore see no correlation between the number of documents opened and success.

Participants' Strategies

Participants were allowed to use any tool accessible to them to solve the tasks. However, all participants exclusively utilized the DDNA Analyzer

to solve Tasks 1 and 2. Figure 7.24 shows how often the two commands of the DDNA Analyzer were used.

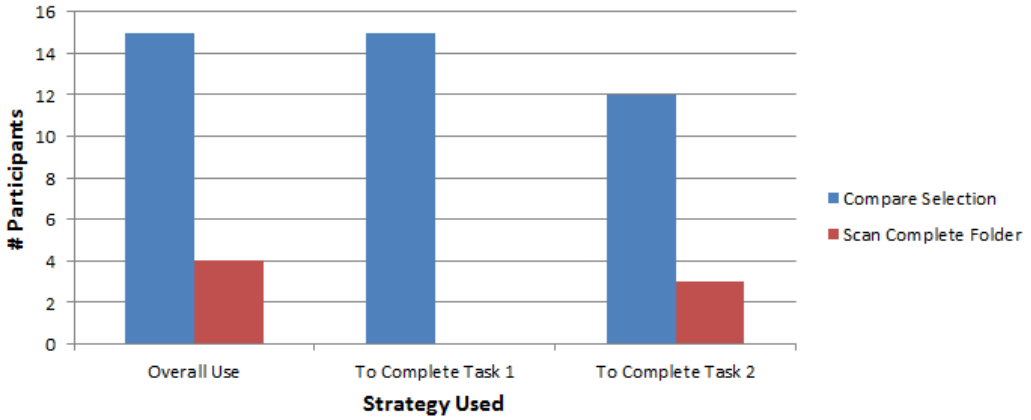


Figure 7.24: Strategies — Control Group

Command: Compare Selection All participants used the `compare selection` command to solve Task 1 and all but three (P22, P27 and P28) participants used the command to solve Task 2. All participants using this command used the `ctrl+a` command to select all content within a document. This selection was then compared against either the Task 1 and 2 folders or sub-folders.

Command: Scan Complete Folder Four participants (P16, P22, P27, and P28) tried to use the `scan complete folder` command to solve Task 1 but failed to do so. Three participants (P22, P27, and P28) used the command to solve Task 2.

Voluntary Participant Feedback

Four participants (P16, P23, P28, and P29) remarked that they would like the vertices to feature the same colors as the inbound edges. They said that this would have increased their confidence in their results.

7.5 Analysis & Discussion

We defined measurements of speed, accuracy, and confidence to explore how content-centered provenance data tracking influenced the result quality of tasks performed by the participants. The following sections analyze the difference between the control and non-control group for these mea-

surements.

7.5.1 Speed & Opened Documents

We measured the speed in which tasks were executed in minutes.

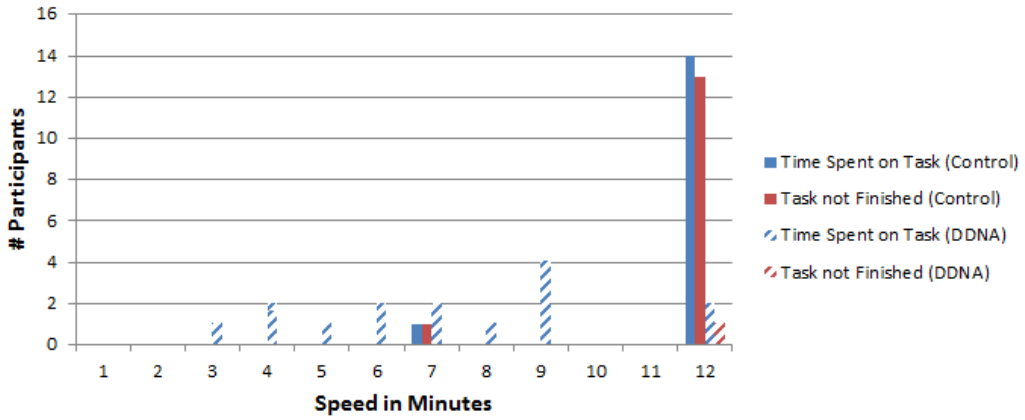


Figure 7.25: Task 1 — Speed Comparison

Figure 7.25 shows a direct comparison of the time that the participants spent on Task 1. The figure shows a sharp decline in time needed for the non-control group. This trend is furthermore supported by the fact that fourteen control group participants failed to finish the task. Only one participant failed to finish the task using the DDNA Analyzer.

We believe the reason for this decline in time needed is a direct result of the strategy change of the participants. Participants that used the DDNA Analyzer did not need to compare content directly or scan the document names for keywords. Instead, they utilized the relationships represented by the DDNA Analyzer to finish the task.

The time participants spent on Task 2 is compared in Figure 7.26. This figure also shows a decline in time needed, although the reduction is not as significant as for Task 1. The arguments for the improvement in Task 1 can also be applied for Task 2. However, part of the second task required the participants to extract information (the contract year) from the folder names or document content. We believe this is the reason that the difference measured for Task 2 is not as significant as for Task 1.

We also observed a sharp decline in the number of documents that participants opened, as shown in Figures 7.27 and 7.28. We believe this is

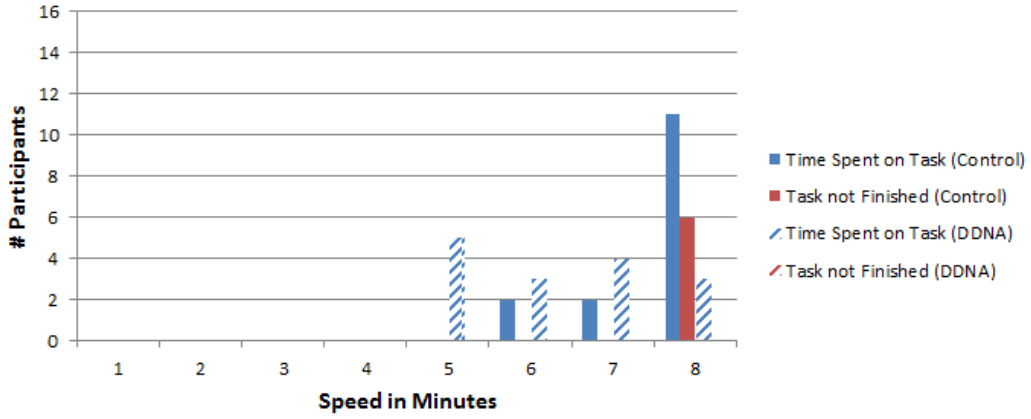


Figure 7.26: Task 2 — Speed Comparison

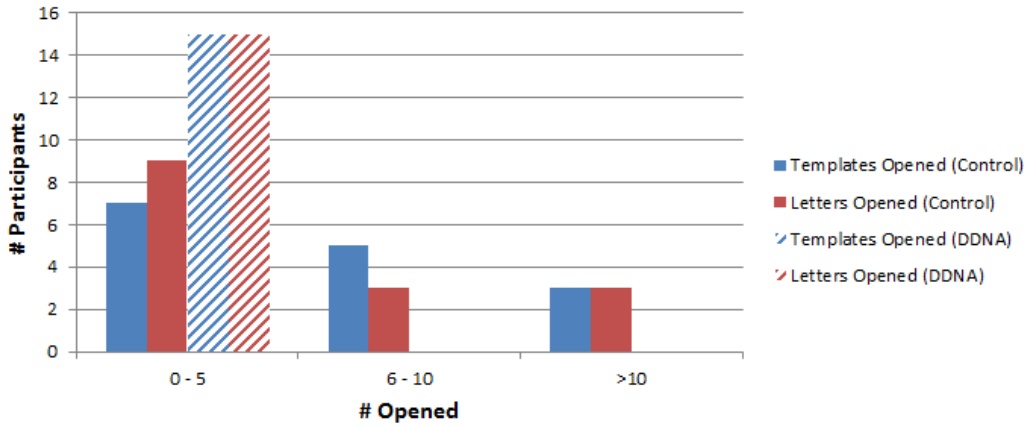


Figure 7.27: Task 1 — Documents Opened Comparison

caused by the fact that participants in the non-control group relied on the information gained by the DDNA-Analyzer, instead of scanning the content of the documents. The decrease in opened documents also improves the overall speed of the participants.

7.5.2 Accuracy & Confidence

We measured the accuracy by sorting the outcomes of the tasks into three categories: fail, partial success and success. Confidence was measured on a Likert Scale from 1 (no confidence) – 5 (full confidence).

The difference in accuracy and confidence between the non-control and control group is shown in Figure 7.29. The figure shows a sharp increase in accuracy, as the majority (13/15) of the non-control group succeeded in

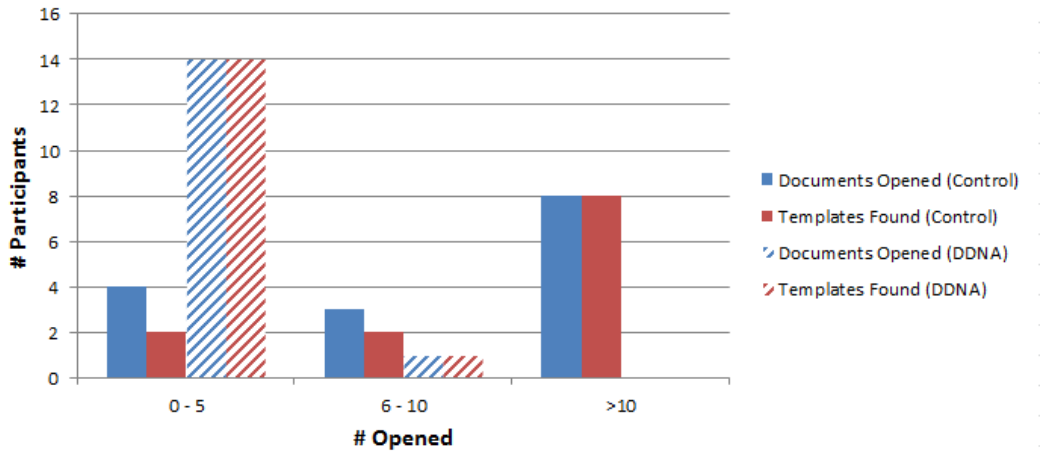


Figure 7.28: Task 2 — Documents Opened Comparison

completing Task 1, whilst the majority (11/15) of the control group failed. We believe the reason for the increased accuracy can be directly attributed to the accuracy of the DDNA Analyzer. As the results shown by the DDNA Analyzer were complete and correct, the participants' results were, too.

We also observed a slight increase in confidence for the participants with partial success. However, as the sample size for partial success was quite small (2 participants in the non-control group), we are unable to draw conclusions from this increase. The difference in confidence between the participants who failed and succeeded can be attributed to the fact that most participants who failed were fully aware of that fact.

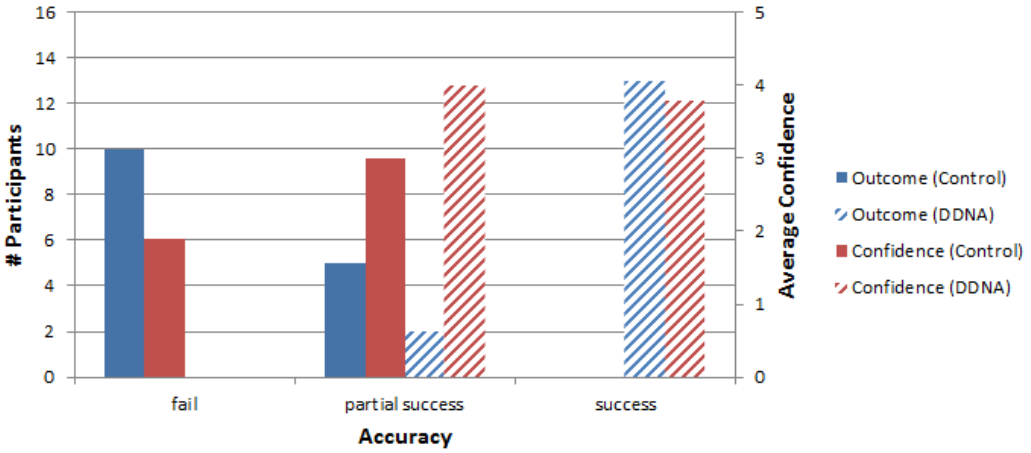


Figure 7.29: Task 1 — Accuracy Comparison

Figure 7.30 shows a comparison of accuracy and confidence for Task

2 between the non-control and control group. We again observe a sharp increase in accuracy between the control (4/15 success) and non-control (13/15 success) group. As for Task 1, the main reason behind this increase is that the results given by the DDNA Analyzer are correct and complete. The difference in confidence is negligible for partial success and success, as participants seemed to be aware of whether or not they completed the task successfully without or with the interface.

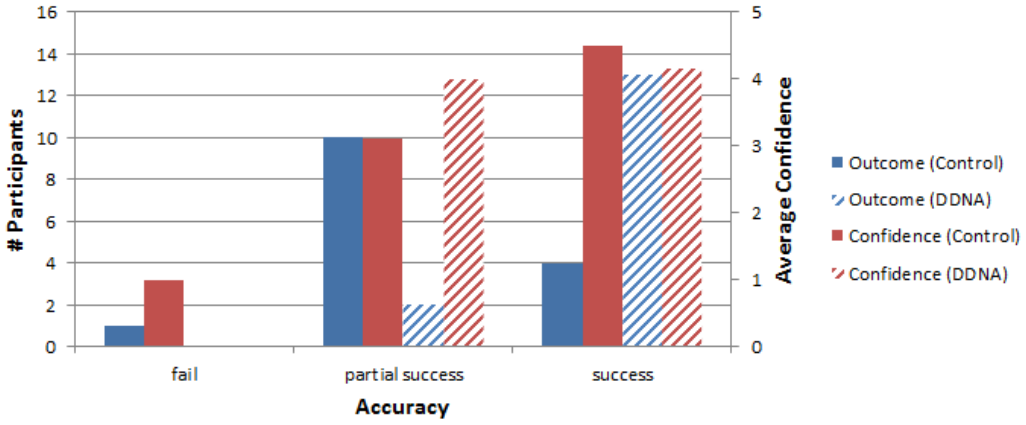


Figure 7.30: Task 2 — Accuracy Comparison

7.6 Summary and Conclusion

In this chapter, we answered the last two research questions:

- How can we measure the effect of using content-centered provenance data tracking?
- Does content-centered provenance data tracking increase the result quality of the tasks identified?

First, we identified the measures of speed, accuracy, and confidence for the result quality of the tasks to answer Research Question 4. We then described how we collected realistic data to create scenarios and tasks fitting our categories. We used these scenarios and tasks to conduct a user study with a control and non-control group. We used the results of this study to answer Research Question 5.

Research Question 4 We identified the measures of speed, accuracy and confidence. Speed is an important measurement as a correct result can be

useless if it is achieved to slow. Accuracy is important because fast results are useless if they are not correct. Finally, confidence is important to avoid users questioning their results, which can lead to unnecessary repetitions of tasks.

Research Question 5 We conducted a user study including real world scenarios and data. We invited fifteen participants to try to solve the scenarios without the use of provenance data supplied by the DDNA Analyzer and fifteen participants that were allowed to use provenance data. We were able to observe a significant increase in speed and accuracy for the participants utilizing provenance data through the DDNA Analyzer. However, the confidence of participants did not significantly differ for either scenario.

8

Summary and Conclusions

This thesis presented a new content-centered approach to provenance data tracking: Document DNA (DDNA). The approach presented was aimed at supporting knowledge workers who are overwhelmed with structuring, maintaining and finding re-used content. Our approach is centered on the following research hypothesis:

> Content-centered provenance data tracking increases the quality of the results of tasks knowledge workers perform when working with digital data.

This chapter summarizes in Section 8.1 the steps taken to investigate our hypothesis and answer the research questions. Section 8.2 follows with a detailed description of the main contributions included in this thesis and Section 8.3 gives detailed answers to the research questions. Section 8.4 discusses the limitations of our approach and Section 8.5 explores ideas for future work, including enhancements to the implemented prototype, propositions for future studies and general ideas resulting from this research. Finally, Section 8.6 concludes this chapter and overall, this thesis.

8.1 Summary

The first chapter introduced our hypothesis and defined five research questions. In Chapter 2, we defined the terms knowledge worker and provenance data as well as task categories performed by knowledge workers. We

identified issues with the current approach (the file-folder system) to handling digital data, but also discussed automated and user-driven metadata annotation systems and how well they addressed the issues. The task categories defined and issues found where then used to develop requirements for a system addressing the issues. The requirements for the system were:

1. Relationship Detection — The system needs to be able to determine if two digital objects are related, i.e., is one originating from the other, or do they share a point of origin?
2. Relationship Metric — The system needs to enable the user to determine the nature of difference between two related digital objects. E.g., how much do they differ in content semantically and syntactically?
3. Distributed — The metadata needs be stored with the content, instead of separately.
4. Automated — The metadata needs to be created automatically and accurately.

In Chapter 3, we discussed provenance data annotation systems to determine if they address the issues found in Chapter 2 by fulfilling the requirements set. The examined systems only partially fulfilled the requirements, with no system addressing (even partially) all requirements. However, the approaches using provenance data were shown to be more promising than all the metadata annotation systems analyzed in Chapter 2.

We verified the issues stated and task categories defined in Chapters 2 and 3 by conducting three exploratory studies which were reported in Chapter 4. The first study was an interview series that confirmed the main issues of knowledge workers using digital data. The second study targeted knowledge workers using document management systems (DMSs), verified that the issues found were not sufficiently addressed by the DMSs, and confirmed the task categories identified. The third exploratory study discovered additional issues related to the introduction of DMS into the workspace of knowledge workers and confirmed the requirements set in Section 2.4.4.

We introduced our DDNA model in Chapter 5, based on the identified requirements and inspired by the DNA of life-forms. The model is based

on Actions and Sessions to create a history of manipulation. That history is used to create the DDNA, which is then used to define relationships between content. These relationships can be used to answer queries about the content, such as finding the newest and oldest versions of content, or branched-off versions of content.

We implemented the DDNA model in a software prototype via the DDNA Tracker (tracking the provenance data) and DDNA Analyzer (visualizing the provenance data), as introduced in Chapter 6. Both prototypes are Microsoft Word Add-Ins that seamlessly blend into the work processes of knowledge workers.

We finally conducted a user study evaluating our concept and our overall hypothesis in Chapter 7. The study used the DDNA Tracker and DDNA Analyzer to evaluate if our concept is addressing the issues found by measuring the result quality of knowledge workers tasks. The knowledge workers were observed performing tasks with and without the DDNA Analyzer. The study used real world scenarios and data gained using the DDNA Tracker. The results of the study showed a significant increase in result quality for the knowledge workers that used the DDNA Analyzer.

8.2 Contributions

This section summarizes the contributions made by this thesis towards the research field of provenance based annotation systems.

8.2.1 Review and Analysis of Metadata Annotation Systems — Requirements

Our analysis of research on metadata annotation systems and provenance data annotation systems uncovers issues shared by all approaches. These issues are vital in defining requirements for a better approach. Our analysis also explored beneficial features of the reviewed approaches. These features are also vital for the design of a better approach.

8.2.2 User Study on Knowledge Workers using Digital Data

This exploratory study verified issues related to knowledge workers using digital data. The study confirms that most issues are related to the re-use of content across documents. The study also confirms that current systems based on a file-folder structure are not suitable anymore as the file-folder structure is causing some of the issues, such as a high amount of time spent on maintaining the file-folder structure.

8.2.3 User Study on Knowledge Workers Using Document Management Systems

This exploratory study contributed in two ways. The first contribution was discovering which task categories are most important to knowledge workers using digital data and examining if DMSs are supporting these task categories. The second contribution of the study was the confirmation DMSs are not addressing the issues found sufficiently.

8.2.4 Case Study on the Introduction of Document Management Systems

The case study uncovered issues related to the introduction of DMS into the work environment of a group of knowledge workers. The study showed that user-saturation and data-saturation are crucial for the success of a DMS. User-saturation is the amount of users actively using the DMS and data-saturation is the amount of data managed by the system versus the amount of data outside of it. The study showed that decentralization and automation are important requirements for a successful system, as they positively relate to user-saturation and data-saturation.

8.2.5 Design of the DDNA Model

The DDNA model contributes a method for decentralized and content-centered provenance data tracking. It was inspired by the DNA of life-forms and shows how the tracking of provenance data can be performed on content level. The design also defined what relationships could be deducted from the gained data and what queries were possible on these relationships. The model fulfills our requirements for a content-centered

provenance data tracking system.

8.2.6 Implementation of the DDNA Prototype

The implementation of the DDNA model as software prototype showed how the DDNA design could be built in an automated and decentralized way. This showed how a metadata annotation system could provide accurate data, without relying on user input and a central management point, such as a web server.

8.2.7 An Evaluation of the Usefulness of Content-Centered Provenance Data in a User Study.

Our evaluation contributes in three ways. The first contribution is identifying how to evaluate the result quality of tasks performed by knowledge workers through measuring speed, accuracy and confidence. Further, we showed how to apply these measurements to the task categories we defined.

The second contribution is two real world scenarios using real word data gained through deploying the DDNA Tracker for a year with a volunteering knowledge worker. The data was confirmed to be accurate by the knowledge worker and the scenarios are based on a use case described by the knowledge worker. The scenarios are therefore realistic and allow for accurate results when measuring the quality of new systems.

The third contribution is the study we conducted using the scenarios and data gained. In this study, we observed both knowledge workers using the DDNA Analyzer and knowledge workers not using it. The knowledge workers were asked to perform two tasks contained in the scenarios, and we measured the result quality following the identified measures. The study contributed by showing a significant increase in result quality of tasks for knowledge workers using the DDNA Analyzer.

8.3 Answers to Research Questions

This section includes the answers to the five research questions.

8.3.1 What Tasks Do Knowledge Workers Perform when Working with Digital Content?

In order to understand how to help knowledge workers using digital data, we needed to understand what it is knowledge workers do. We began answering this question by defining what the term knowledge worker was and defining what constituted digital data in this research. Next, we reviewed research on knowledge workers and extracted roles of knowledge workers and task categories associated with these roles.

We reduced the original range of task categories by filtering for tasks that are performed using digital data and tasks that are connected to the majority of roles. The resulting task categories were: *Information Search*, *Information Organization*, *(Co)-Authoring Information* and *Information Dissemination*.

To verify that the selection was a representation of the tasks real world knowledge workers perform, we conducted exploratory studies. The studies' results confirmed the initial task selection and added *Versioning* as a composite task. We also discovered that these tasks were mainly performed on re-used content.

8.3.2 What Are the Issues with Current Systems Aimed to Support Knowledge Work?

After understanding what tasks knowledge workers are performing, we reviewed current systems designed to support these tasks to identify what issues are related to these systems. These systems were identified to be metadata annotation systems. We began with the basic metadata annotation system available to all knowledge workers, the file-folder system, but also reviewed metadata annotation systems developed to replace or complement the file-folder system.

We found that the most useful metadata to annotate is provenance data and therefore conducted a separate review of provenance focused annotation systems. We found that the three most important issues with all systems reviewed were: file-centricity, reliance on a central managing point and the reliance of manual user input for accuracy.

Issue 1: File-Centricity All systems reviewed tracked metadata on file level. This resulted in inaccurate or even conflicting metadata because files can contain different pieces of content that require different metadata annotations. Also, the metadata annotations of specific content are lost if only that content is transferred to another file, instead of transferring the whole file.

Issue 2: Reliance on Central Managing Point All reviewed systems stored the metadata separated from the content and most of the systems relied on a central database or server to manage the metadata. This is an issue, because content leaving the reach of the metadata annotation system loses the annotation. Furthermore, metadata was only tracked within the reach of the central managing point of the metadata annotation system, which can result in missing crucial metadata, or false metadata.

Issue 3: Reliance on Manual Input We reviewed automated and user-driven systems. The automated systems could not guarantee accurate metadata. The manual systems were able to produce accurate metadata, but had limited scalability because of the needed user input.

To confirm the identified issues, we conducted three user studies which explored issues of knowledge workers using digital data. The first study confirmed that knowledge workers using the file-folder structure struggled with maintaining, synchronizing, versioning and updating their digital content. This was mostly caused by Issues 1 and 3, as the file-folder system is file-centered and reliant on user input.

We then verified that current DMSs used are not sufficiently addressing these issues by conducting a second study targeting knowledge workers using DMSs. The studies' results confirmed that the DMSs were partially addressing Issues 1 and 3, but failing to address Issue 2, which negated the partial successes achieved for Issues 1 and 3.

The third user study confirmed that introducing DMSs can cause issues, as DMSs are prone to fail for two reasons: low user saturation and low content saturation. This is again a confirmation of Issues 2 and 3, because the dependance of active user participation for the central DMS resulted in low user saturation. Low user saturation caused low content saturation, as users are required to input and maintain the content.

8.3.3 How Can Content-Centered Provenance Data Tracking Be Implemented?

The issues found in the reviews and user studies resulted in the following requirements.

1. Relationship Detection — The system needs to be able to determine if two digital objects are related, i.e., is one originating from the other, or do they share a point of origin?
2. Relationship Metric — The system needs to enable the user to determine the nature of difference between two related digital objects. E.g., how much do they differ in content semantically and syntactically?
3. Distributed — The metadata needs be stored with the content, instead of separately.
4. Automated — The metadata needs to be created automatically and accurately.

Requirements 1 and 2 address Issue 1, Requirement 3 addresses Issue 2 and Requirement 4 addresses Issue 3. To realize these requirements, we firstly introduced the DDNA design. The DDNA design utilizes the concept of Actions and Sessions to track manipulations to content and clearly distinguishes the content from the containing file — the document. The design also specifies how the information gained through tracking Actions and Sessions can be translated into ancestor/descendant and sibling relationships and how these relationships can be used for queries. The design fulfilled Requirements 1 and 2.

We then developed the DDNA design and implemented it as two Microsoft Word add-ins, the DDNA Tracker and DDNA Analyzer. The DDNA Tracker implements the action and session concepts through automated tracking of content manipulation in documents, including copy and paste events between documents. The information gained is stored within the document holding the content. The DDNA Tracker therefore fulfilled Requirements 3 and 4. The DDNA Analyzer implemented an interface to allow users to use queries (as specified in the DDNA design) on the newly acquired provenance information.

8.3.4 How Can we Measure the Effect of Using Content-Centered Provenance Data Tracking?

Whether or not the DDNA Tracker and DDNA Analyzer addressed issues of knowledge workers can be detected by measuring the result quality of tasks knowledge workers are performing. To be able to evaluate the result quality of tasks, we needed to determine what values were important for the success of the tasks. We identified speed, accuracy and confidence as the three main measurements that determined whether or not a task was completed successfully.

Speed: No matter how accurate the result of a task is, if the result arrives to late, it might be useless. Examples of this are time critical tasks like searching for important facts before a presentation.

Accuracy: The accuracy of task results is a major factor for the success of a task. Results that are gained quickly are useless if the results are inaccurate.

Confidence Whilst not as important as speed and accuracy, confidence can still be a factor for task result quality. Knowledge workers who have a low confidence in the accuracy of a task they have performed might decide to perform a task with a good result again, therefore wasting time. Even worse is a high confidence in a task result that has very low accuracy, resulting in the knowledge worker using a false result.

8.3.5 Does Content-Centered Provenance Data Tracking Increase the Result Quality of the Tasks Identified?

With the finished definitions on measurements for the identified tasks, we needed to design a study to test the result quality with and without the provenance data supplied by the DDNA Tracker and DDNA Analyzer. We conducted a year long data gathering trial with a participating knowledge worker using the DDNA Tracker, to be able to use real world data and scenarios for the study.

We used the gained scenario and data to conduct a user study with two participant groups: one group using the DDNA Analyzer and one (control) group without, conducting two tasks. The results of the study

showed significant increases in speed and accuracy for the observed tasks whilst results for confidence were indecisive. This proves that the use of content-centered provenance data is a valid approach for addressing the issues knowledge workers have.

8.4 Limitations

This section discusses the limitations of model and software. Both were sufficient to explore our hypothesis, but both also have limitations caused by the limited scope of this research.

8.4.1 Model

Our concept's main limitation is the focus on text. The atomic actions are focused on text manipulations and text transfer. However, other types of content feature other manipulation methods. For example, changing the color spectrum of an image. Our design is therefore only partially suited to track content other than text.

8.4.2 Software

Our software has three limitations: detecting all types of content transfer, tracking changes outside of Microsoft Word, and missing Privacy and Integrity protection.

Content Transfer The first limitation is related to the detection of content transfer. Our concept is based on tracking such transfers to be able to supply accurate provenance data. However, content can be transferred outside the copy and paste technique employed in most scenarios. For example, a knowledge worker could decide to manually copy text information by re-typing the text into the new document. Or a knowledge worker could decide to redraw a figure, while looking at a printout of the figure. Such content transfers are an integral part of the history of content, but cannot be detected by our system.

Tracking Depth The second limitation is caused by the need for decentralization. We need to be able to attach the DDNA to the content, in order to avoid a central managing point. However, not every content container allows adding custom metadata. We also rely on the content editor to be able

to seamlessly track the evolution of the content described by the DDNA. If content is edited with differing editors, there needs to be only one editor not supporting the DDNA to lose evolution information, or worse, the whole DDNA.

Privacy & Integrity The third limitation is caused by the prototypical nature of our implementation. We have not implemented an option to delete or reset the DDNA of content, which is an option that would allow users to preserve privacy before passing content on to other users. Secondly, the current implementation of the DDNA uses the custom XML part of the *.docx* structure. This part is easily accessible by users and could therefore be tampered with and therefore cripple the integrity of the DDNA.

8.5 Future Work

This section discusses in which ways our concept and implementation could be refined in the future. We also discuss possible future studies and further ideas resulting from this research.

8.5.1 DDNA Model

Relation Strength — DNA The DDNA Model addresses the quantitative difference between two pieces of content by determining the numbers of versions between the two pieces. We propose the introduction of a second, qualitative measurement to allow for a better definition of relation strength. A qualitative measurement could be achieved via a bag of words approach, where a bag of words is created to represent the most important words used in a piece of content. By comparing two different bags of words, we can determine how much two pieces of content differ semantically. However, other approaches to comparing document similarity (Shivakumar and Garcia-Molina, 1999; Stein, 2007; Huang, 2011) are also promising.

Content Covered The current DDNA Model supports provenance tracking in text-based digitasl objects. Provenance of other types of content, such as images, audio or video have not been addressed so far. A first step to support other content types requires an analysis of the short commings of the current model with regard to those content types. In a second step,

the DDNA model needs to be adapted to support an extended Document concept and new Actions.

8.5.2 DDNA Tracker

There are two areas where improvements to the DDNA Tracker are possible: relation strength, tracking depth and content covered.

Tracking Depth The DDNA Tracker is able to track all content transfers and manipulations executed in Microsoft Word. As discussed in our limitations, we needed to be able to track manipulations and transfers executed using other editors, too. This is because missing transfers or manipulations lead to less accurate data. One way to achieve a better tracking depth would be to move the tracking to the operating system level. Additionally, we need a universal way to attach the DDNA to content for all content containers.

Content Covered The DDNA Tracker's focus is content included in Microsoft Word documents, which are mainly text. We believe that the DDNA Tracker needs to be able to also track other types of content, such as images, audio or video. The DDNA Tracker is able to track such content's placement and transfer between documents, but not the changes made to the content. Since content can be combined in every imaginable way, the DDNA Tracker needs to be able to track all types of content equally. We therefore think that we need to further develop our software to be able to track changes made to non-text content.

8.5.3 DDNA Analyzer

We see three main areas for improvement for the DDNA Analyzer: interface detail, interface flexibility and alternative interfaces.

Interface Detail: The DDNA Analyzer provides a basic graph-based interface illustrating the relations between documents. The level of detail can be improved by adding information about the ratio of different content pieces in a document. Another possibility to add detail is to highlight content in the color it is represented by in the interface, when opening documents.

Interface Flexibility: The DDNA Analyzer allows users to realign the vertices in the current interface via drag and drop. However, more flexibility

could be provided by allowing users to create sub-graphs based on vertex selection on the fly. Flexibility could also be added by allowing users to filter the graph based on search parameters like creation date or file path.

Alternative Interfaces: While we achieved good results using a graph representation for the detected content relationships, we believe other methods are also promising (Hetzler et al., 1998). Instead of a graph, the relationships detected via the DDNA could be displayed as three dimensional map with a tangible motion controlled interface. However, first we need to research clustering methods for such a map.

8.5.4 Studies

This section discusses future fields of study where our approach may be useful.

Collaborative Work: The final study focused on a scenario where single users were introduced to an existing data structure and tasked to use the data structure to gain information. While this is a valid scenario that often happens in the real world, we also need to consider other scenarios. We propose a long term study on collaborative groups of knowledge workers. Although such a study has been planned, for practical and logistical reasons it was not possible to be conducted. We believe that the issues we identified in this thesis, such as versioning or synchronizing, gain increased importance in a collaborative environment. This is because more users working on the same content base means more errors made when versioning that content. Additionally, more instances of synchronizing the content occur. It is therefore very likely that our DDNA concept would also improve the result quality of collaborative tasks of knowledge workers, which should be explored in a suitable large–scale study.

Digital Library Visualization: Detecting and visualizing content and citation relations in digital libraries (Chen, 1999) is another prospective application for the DDNA. However, such a study would depend either on a lengthy data collection phase or a retroactive analysis to detect all existing relations.

Work Flow Detection: Provenance has been used before to determine important processes in a work environment (Shen et al., 2009). We believe the

DDNA could be used in a number of ways to improve work flows, such as determining important documents and common document relations. We propose to study the accuracy of work flow detection supported by the DDNA.

Trust: Ko et al. (2011b) stated that trust is an important factor for users managing data in a cloud. We believe the DDNA model and software could be applied to measure the difference in trust between users using a cloud application for documents like GoogleDocs (Google, 2010) with a DDNA modification and users using the standard cloud application.

8.5.5 Further Ideas

This section includes further ideas for future research related to the DDNA.

Benchmarking: Many research fields like Machine Learning or Formal Methods have widely accepted benchmarks to evaluate new algorithms or systems. Such benchmarks do not exist to evaluate interfaces or full systems for knowledge workers. HCI researchers are usually tailoring their evaluation towards the system they propose. This hinders comparing different approaches to similar issues, as the design of the evaluation is a major contributing factor to the outcome. We therefore suggest the introduction of a base set of realistic tasks using realistic data that is openly available for HCI researchers to compare the validity of their systems. The DDNA could be used to detect the data for such a base set.

Plagiarism: Detecting plagiarism is another application for the DDNA. Detecting suitable search strings (Culwin and Child, 2010), semantic similarity (Tsatsaronis et al., 2010) and near-replicas (Shivakumar and Garcia-Molina, 1999) are all methods used to help plagiarism detection, and are all methods we believe could be supported by the DDNA. However, the DDNA is currently not implemented robustly enough to withstand malicious user manipulation, which is an issue that needs to be addressed before moving further in that direction.

Information Retrieval: Information Retrieval measures focus on how adequate content is in relation to a specific search query, while our approach aims to measure the provenance of content, i.e.: how is content related and what is the origin of content. The work described in this thesis could be

combined with Information Retrieval approaches to search for both specific content and document provenance relations.

8.6 Conclusion

This thesis has shown that content-centered provenance data tracking can improve the result quality of knowledge workers tasks. It introduced a detailed review of metadata annotation systems and three studies identifying current issues knowledge workers using digital content have. We introduced a design and implementation for a distributed and content-centered provenance data tracking system addressing these issues. Finally, we conducted a study using real world data and scenarios, showing that the introduced system has increased the result quality of knowledge workers significantly.

References

Apache Software Foundation (2014). Subversion. http://subversion.apache.org/. Last accessed September 2014.

Bagga, A., Baldwin, B. (1998). Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*, pp. 79–85. Association for Computational Linguistics.

Bao, X., Herlocker, J. L., Dietterich, T. G. (2006). Fewer clicks and less frustration: reducing the cost of reaching the right folder. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, pp. 178–185. ACM.

Barga, R. S., Simmhan, Y. L., Chinthaka, E., Sahoo, S. S., Jackson, J., Araujo, N. (2010). Provenance for scientific workflows towards reproducible research. *IEEE Data Engineering Bulletin*, *33*(3), 50–58.

Barreau, D., Nardi, B. A. (1995). Finding and reminding: file organization from the desktop. *ACM SigChi Bulletin*, *27*(3), 39–43.

Black, I. (2013). NSA spying scandal: what we have learned. *The Guardian*. Last accessed September 2014. http://www.theguardian.com/world/2013/jun/10/nsa-spying-scandal-what-we-have-learned

Boese, E. S., Howe, A. E. (2005). Effects of web document evolution on genre classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pp. 632–639. ACM.

Briggs, P., Smyth, B. (2008). Provenance, trust, and sharing in peer-to-peer case-based web search. In *Advances in case-based reasoning*, pp. 89–103. Springer.

Bush, V. (1945). As we may think. *The Atlantic Monthly*, *176*(1), 101–108.

Chen, C. (1999). Visualising semantic spaces and author co-citation networks in digital libraries. *Information Processing & Management*, *35*(3), 401–420.

Chirita, P., Gavriloaie, R., Ghita, S., Nejdl, W., Paiu, R. (2005). Activity based metadata for semantic desktop search. In Gómez-Pérez, A., Euzenat, J. (Eds.), *The Semantic Web: Research and Applications*, vol. 3532 of *Lecture Notes in Computer Science*, pp. 439–454. Springer. ISBN 978-3-540-26124-7.

Cortada, J. (1998). *Rise of the knowledge worker*. Taylor & Francis Ebooks.

de la Cruz, F., Davies, J. (2000). Horizontal gene transfer and the origin of species: lessons from bacteria. *Trends in microbiology*, *8*(3), 128–133.

Culwin, F., Child, M. (2010). Optimising and automating the choice of search strings when investigating possible plagiarism. In *Proceedings of 4th International Plagiarism Conference*.

De Chiara, R., Erra, U., Scarano, V. (2003). VennFS: A Venn-diagram file manager. In *7th International Conference on Information Visualisation*, pp. 120–120. IEEE.

Di Iorio, A., Vitali, F. (2003). A Xanalogical collaborative editing environment. In *Proceedings of the Second International Workshop of Web Document Analysis*, pp. 47–50.

Dighe, A., Hinze, A. (2012). Human-centred workplace: Re-finding physical documents in an office environment. In *Proceedings of the 13th International Conference of the NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction*, pp. 9–16. ACM.

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J. (2004). Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, pp. 652–659. ACM.

Dragunov, A. N., Dietterich, T. G., Johnsrude, K., McLaughlin, M., Li, L., Herlocker, J. L. (2005). TaskTracer: a desktop environment to support multi-tasking knowledge workers. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, pp. 75–82. ACM.

Drucker, P. (1964). Knowledge worker: new target for management. *Christian Science Monitor*, *10*.

Fallin, K., Wyvill, B. (2003). Entropy: Managing data in an electronic world. Undergraduate research project, University of Calgary.

Fertig, S., Freeman, E., Gelernter, D. (1996a). "finding and reminding" reconsidered. *ACM SIGCHI Bulletin*, *28*(1), 66–69.

Fertig, S., Freeman, E., Gelernter, D. (1996b). Lifestreams: an alternative to the desktop metaphor. In *Conference Companion on Human Factors in Computing Systems*, pp. 410–411. ACM.

Gantz, J., Reinsel, D. (2012). The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the Future*. Last accessed September 2014.
http://www.emc.com/collateral/analyst-reports/idc-digital-universe-united-states.pdf

Ghorashi, S., Jensen, C. (2013). The Leyline: A comparative approach to designing a graphical provenance-based search ui. In *46th Hawaii International Conference on System Sciences*, pp. 1630–1639. IEEE.

Golbeck, J. (2006). Combining provenance with trust in social networks for semantic web content filtering. In *Provenance and Annotation of Data*, pp. 101–108. Springer.

Google (2010). Googledocs. https://docs.google.com/. Last accessed September 2014.

Grevstad, E. (2003). Scopeware vision review: Bring clutter and chaos (and maybe your pc) to their knees. *Winplanet Software Reviews*. Last accessed September 2014.
http://www.smallbusinesscomputing.com/biztools/article.php/2193081/Scopeware-Vision-Review.htm

Hamano, J., Torvalds, L. (2014). Git. http://git-scm.com/. Last accessed September 2014.

Hetzler, E., Harris, W. M., Havre, S., Whitney, P. (1998). Visualizing the full spectrum of document relationships. *Advances in Knowledge Organization*, *6*, 167–174.

Hopkins, I., Vassileva, J. (2005). Beyond keywords and hierarchies. *Journal of Digital Information Management*, *3*(2), 139.

Huang, L. (2011). *Concept-based text clustering*. Ph.D. thesis, Department of Computer Science of the University of Waikato.

Jensen, C., Lonsdale, H., Wynn, E., Cao, J., Slater, M., Dietterich, T. G. (2010). The life and times of files and information: a study of desktop provenance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 767–776. ACM.

Jervis, M., Masoodian, M. (2013). Visualization of physical library shelves to facilitate collection management and retrieval. In *Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pp. 133–138. ACM.

Karlson, A. K., Smith, G., Lee, B. (2011). Which version is this?: improving the desktop experience within a copy-aware computing ecosystem. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2669–2678. ACM.

Karypidis, A., Lalis, S. (2006). Omnistore: A system for ubiquitous personal storage management. In *4th IEEE International Conference on Pervasive Computing and Communications*, pp. 136–147. IEEE.

Karypidis, A., Lalis, S. (2007). Automated context aggregation and file annotation for PAN-based computing. *Personal and Ubiquitous Computing*, *11*(1), 33–44.

Keiser, V. L. (2009). Evaluating online text classification algorithms for email prediction in tasktracer. In *Conference on Email and Anti-Spam*. CEAS.

Ko, R. K., Jagadpramana, P., Lee, B. S. (2011a). Flogger: A file-centric logger for monitoring file access and transfers within cloud computing environments. In *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 765–771. IEEE.

Ko, R. K., Jagadpramana, P., Mowbray, M., Pearson, S., Kirchberg, M., Liang, Q., Lee, B. S. (2011b). TrustCloud: A framework for accountability and trust in cloud computing. In *IEEE World Congress on Services (SERVICES)*, pp. 584–588. IEEE.

Lamport, L. (2011). Latex. http://latex-project.org/. Last accessed September 2014.

Lesk, A. (2013). *Introduction to bioinformatics*. Oxford University Press.

Lettkeman, A. T., Stumpf, S., Irvine, J., Herlocker, J. (2006). Predicting task-specific webpages for revisiting. In *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, pp. 1369–1374. MIT Press.

Mackall, M. (2014). Mercurial. http://mercurial.selenic.com/. Last accessed September 2014.

Mesnier, M., Thereska, E., Ganger, G. R., Ellard, D., Seltzer, M. (2004). File classification in self-* storage systems. In *Proceedings of the International Conference on Autonomic Computing*, pp. 44–51. IEEE.

Microsoft (2011). Windows 7 SP1 (6.1.7601). http://support.microsoft.com/product/windows/windows-7/. Last accessed September 2014.

Microsoft (2014). Mircosoft Office 15.0.4623.1003. http://office.microsoft.com/en-us/word/. Last accessed September 2014.

Nelson, T. H. (1999). Xanalogical structure, needed now more than ever: parallel documents, deep links to content, deep versioning, and deep re-use. *ACM Computing Surveys (CSUR)*, *31*(4es), 33.

Oracle Corporation (2011). Open Office. https://www.openoffice.org/. Last accessed September 2014.

PantheR (2014). Graphx for .net 2.2. http://graphx.codeplex.com/. Last accessed July 2014.

Papantoniou, B. (2014). Cognitive ergonomics. https://www.interaction-design.org/encyclopedia/cognitive_ergonomics.html. Last accessed July 2015.

pelikhan (2011). Quickgraph 3.6. https://quickgraph.codeplex.com/. Last accessed September 2014.

RealWire (2013). Perforce study reveals that 83% of knowledge workers lose time to document versioning issues each day. http://cmsreport.com/articles/perforce-study-reveals-that-83-of-knowledge-workers-lose-time-to-document-versioning-issues-each-day-5273. Last accessed September 2014.

Reinhardt, W., Schmidt, B., Sloep, P., Drachsler, H. (2011). Knowledge worker roles and actions — results of two empirical studies. *Knowledge and Process Management*, *18*(3), 150–174.

Satoh, K., Okumura, A. (1999). Documentation know-how sharing by automatic process tracking. In *Proceedings of the 4th International Conference on Intelligent User Interfaces*, pp. 49–56. ACM.

Schibille, N., Marii, F., Rehren, T. (2008). Characterization and provenance of late antique window glass from the Petra Church in Jordan*. *Archaeometry*, *50*(4), 627–642.

Schleimer, S., Wilkerson, D. S., Aiken, A. (2003). Winnowing: Local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pp. 76–85. ACM.

Schütte, A. A. (1998). *Patina: layering a history-of-use on digital objects*. Master's thesis, Massachusetts Institute of Technology.

Sellen, A. J., Harper, R. H. (2003). *The Myth of the Paperless Office*. The MIT Press, 1st edn.

Shadbolt, N., Hall, W., Berners-Lee, T. (2006). The semantic web revisited. *IEEE Intelligent Systems*, *21*(3), 96–101.

Shen, J., Fitzhenry, E., Dietterich, T. G. (2009). Discovering frequent work procedures from resource connections. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pp. 277–286. ACM.

Shen, J., Li, L., Dietterich, T. G. (2007). Real-time detection of task switches of desktop users. In *The International Joint Conference on Artificial Intelligence*, vol. 7, pp. 2868–2873. IJCAI.

Shenk, D. (1998). *Data smog: Surviving the information glut*. Harper San Francisco.

Shepherd, A. (2015). *Evaluation of Human Work*. CRC Press, 4th edn.

Shivakumar, N., Garcia-Molina, H. (1999). Finding near-replicas of documents on the web. In *The World Wide Web And Databases*, pp. 204–212. Springer.

Signer, B. (2010). What is wrong with digital documents? A conceptual model for structural cross-media content composition and reuse. In *Conceptual Modeling — ER 2010*, pp. 391–404. Springer.

Simmhan, Y. L., Plale, B., Gannon, D. (2005). A survey of data provenance in e-science. *SIGMOD Rec.*, *34*(3), 31–36. Last accessed September 2014.

Soules, C. A., Ganger, G. R. (2003). Why can't I find my files?: New methods for automating attribute assignment. In *Workshop on Hot Topics in Operating Systems (HotOS)*, pp. 115–120. USENIX.

Stein, B. (2007). Principles of hash-based text retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 527–534. ACM.

Sugiyama, K., Misue, K. (1991). Visualization of structural information: Automatic drawing of compound digraphs. *IEEE Transactions on Systems, Man and Cybernetics*, *21*(4), 876–892.

Svensson, M. (2009). Contextual metadata in practice. In *First International Conference on Advances in Multimedia (MMEDIA'09)*, pp. 12–17. IEEE.

Tannen, V. (2008). Provenance for database transformations. In *Provenance and Annotation of Data and Processes*, pp. 1–1. Springer.

The CVS Team (2008). Concurrent versions system. http://www.nongnu.org/cvs/. Last accessed September 2014.

Tsatsaronis, G., Varlamis, I., Giannakoulopoulos, A., Kanellopoulos, N. (2010). Identifying free text plagiarism based on semantic similarity. In *Proceedings of the 4th International Plagiarism Conference*.

Xu, Z., Karlsson, M., Tang, C., Karamanolis, C. T. (2003). Towards a semantic-aware file store. In *Workshop on Hot Topics in Operating Systems (HotOS)*, pp. 181–187. USENIX.

Zhang, J., Jagadish, H. (2013). Revision provenance in text documents of asynchronous collaboration. In *IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 889–900. IEEE.

A

Appendix - User Studies

A.1 Exploratory Study — Interviews

This section contains all related material for the first exploratory study. The ethics approval letter is shown in Figure A.1. The participant information sheet for the first study is shown in Figures A.2 & A.3. The interview guideline for the first study is shown in Figure A.4.

Faculty of Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton 3240
New Zealand

Phone +64 7 838 4322
www.scms.waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

COPY

7 December, 2011

Michael Rinck
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Michael

Request for approval to conduct a study involving human participants for your PhD Research Project.

I have considered your request to conduct interviews with University of Waikato staff for your research project "Connecting Information: Detecting and Tracing Document Evolution".

The procedure described in your request is acceptable. I note your statements that confidentiality and participant anonymity will be strictly maintained, that all data collected will be anonymized and no names or other identifying characteristics will be stated in the final or any other reports. Once analyzed and summarized, the original notes will be destroyed.

The research participants' Information Sheet, and Research Consent comply with the requirements of the University's human research ethics policies and procedures.

I therefore approve your application to perform the user study.

Yours sincerely,

Masood Masoodian
Human Research Ethics Committee
Faculty of Computing and Mathematical Sciences

Figure A.1: Study 1 Ethics Approval Letter

Participant Information Sheet

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Ethics Committee, Faculty of Computing and Mathematical Sciences

Project Title
Connecting Information: Detecting and Tracing Document Evolution.

Purpose
This research is conducted as partial requirement for the Ph.D. thesis of Michael Rinck. This project requires the researcher to choose a topic and conduct research on the topic through using surveys or interviews or a combination of the two techniques.

What is this research project about?
This project aims to detect and trace the evolution of content. This means that we want to capture the process of content editing and attach the gained information directly to the edited content. This information can then be used to detect the path content took before being used in the document at hand. Users can use this information to discover documents with similar content, in other words related documents.
As a first step we would like to interview users about their means to keep track of this relations at the moment, for example how they handle versions of documents when collaborating.

What will you have to do and how long will it take?
You will be asked to participate in an interview at your workplace. You might be asked to show the researcher examples of the techniques you use to store data and keep track of content, e.g. document versions. The interview will take no longer than 15 minutes.

What will happen to the information collected?
The information collected will be used by the researcher to substantiate the need to keep track of information about content evolution. It is possible that articles and presentations may be the outcome of the research. Only the researcher and supervisor Annika Hinze will be privy to the notes and the paper written. Afterwards, notes and documents will be destroyed. The researcher will keep transcriptions of the interviews, these transcriptions will be destroyed as soon as analysis is completed. No participants will be named in the publications and every effort will be made to disguise their identity.

Declaration to participants
If you take part in the study, you have the right to:

- Refuse to answer any particular question, and to withdraw from the study before the first of August 2012
- Ask any further questions about the study that occurs to you during your participation.
- Be given access to a summary of findings from the study when it is concluded.

Who's responsible?
If you have any questions or concerns about the project, either now or in the future, please feel free to contact either:

Researcher:
Michael Rinck
Mr97@students.waikato.ac.nz

Supervisor:
Annika Hinze
hinze@cs.waikato.ac.nz

Figure A.2: Study 1 Participant Information — First Page

Research Consent Form

Ethics Committee, Faculty of Computing and Mathematical Sciences

Connecting Information: Detecting and Tracing Document Evolution

Consent Form for Participants

I have read the **Participant Information Sheet** for this study and have had the details of the study explained to me. My questions about the study have been answered to my satisfaction, and I understand that I may ask further questions at any time.

I also understand that I am free to withdraw from the study before the first of August 2012, or to decline to answer any particular questions in the study. I understand I can withdraw any information I have provided up until the researcher has commenced analysis on my data. I agree to provide information to the researchers under the conditions of confidentiality set out on the **Participant Information Sheet**.

I agree to participate in this study under the conditions set out in the **Participant Information Sheet**.

Signed: _____

Name: _____

Date: _____

Researcher's Name and contact information: Michael Rinck
mr97@students.waikato.ac.nz

Supervisor's Name and contact information: Annika Hinze
hinze@cs.waikato.ac.nz

Figure A.3: Study 1 Participant Information — Second Page

Participant Information

Participant ID:

Age: Gender:

Questions

Years of professional document usage:

Most used document editor:

Do your documents contain reused content (1-5, 5 is very likely):

Do you often want to re find sources of content (1-5):

How do you organize your Documents (when collaborating):

What problems do you encounter when working with content?:

Figure A.4: Study 1 — Interview Guideline

A.2 Exploratory Study — DMS Questionnaire

This section contains all related material for the second exploratory study. The ethics approval letter is shown in Figure A.5. The participant information sheet for the second study is shown in Figures A.6 & A.7. The online questionnaire for the third study is shown in Figures A.8, A.9, A.10, A.11 & A.12.

Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton
New Zealand

Phone +64 7 838 4021
www.scms.waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

5 November 2013

Michael Rinck
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Michael

Request for approval to conduct a research evaluation involving human participants

I have considered your request to carry out a study for your PhD research project *Connecting Information: Detecting and Tracing Document Evolution* to be conducted by a 30 minute interview with staff in corporate companies.

The interview will consist of several questions about the software participants use at work and about their work practices in general. The information collected will be used to substantiate the need to keep track of information about content evolution.

Questionnaires will not request names of participants therefore any resulting articles and presentations will not contain any personal data and their identities will be kept anonymous. Collected information will be kept in note form or on a password protected computer. Once analyzed and summarized the original notes will be stored in the FCMS data archive till the end of 2018 then destroyed.

The procedure described in your request is acceptable.

The research participants' information sheet, consent form and questionnaire meet the requirements of the University's human research ethics policies and procedures.

I therefore approve your application to perform the evaluation.

Yours sincerely,

Michael Mayo
Human Research Ethics Committee
School of Computing and Mathematical Sciences

Figure A.5: Interview Series Ethics Approval Letter

Connecting Information: Detecting and Tracing Document Evolution

* Required

Participant Information Sheet

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Ethics Committee, Faculty of Computing and Mathematical Sciences

Project Title
Connecting Information: Detecting and Tracing Document Evolution.

Purpose
This research is conducted as partial requirement for the Ph.D. thesis of Michael Rinck.

What is this research project about?
This project aims to detect and trace the evolution of content (text in word files). This means that we want to capture the process of content editing and attach the gained information directly to the edited content (the word files). This information can then be used to detect the path content took before being used in the document at hand. Users can use this information to discover documents with similar content, in other words related documents.

What will you have to do and how long will it take?
You will be asked to participate in a survey that will take no more than 10 minutes. The survey will contain several questions about the software you use at work and about your work practices in general. You have the right to refuse to answer any of the questions or stop the survey at any time during the process.

What will happen to the information collected?
The information collected will be used by the researcher to substantiate the need to keep track of information about content evolution. It is possible that articles and presentations may be the outcome of the research. The researcher will keep notes of the interviews. Only the researcher and supervisor Annika Hinze will be privy to the notes. All collected data will be kept in note form, locked in a drawer in G.206 or stored on a password protected PC there. Once analyzed and summarized the original notes and data will be stored in the FCMS data archive till the end of 2018 and then be destroyed. No participants will be named in the publications and every effort will be made to disguise their identity, particularly through the use of ID numbers instead of names.

Declaration to participants
If you take part in the study, you have the right to:

- Refuse to answer any particular question, and to withdraw from the study before the first of March 2014
- Ask any further questions about the study that occurs to you during your participation.
- Be given access to a summary of findings from the study when it is concluded.

Who's responsible?
If you have any questions or concerns about the project, either now or in the future, please feel free to contact either:
Researcher:
Michael Rinck
Mr97@students.waikato.ac.nz
Supervisor:
Annika Hinze
hinze@cs.waikato.ac.nz

Figure A.6: Study 2 Participant Information — First Page

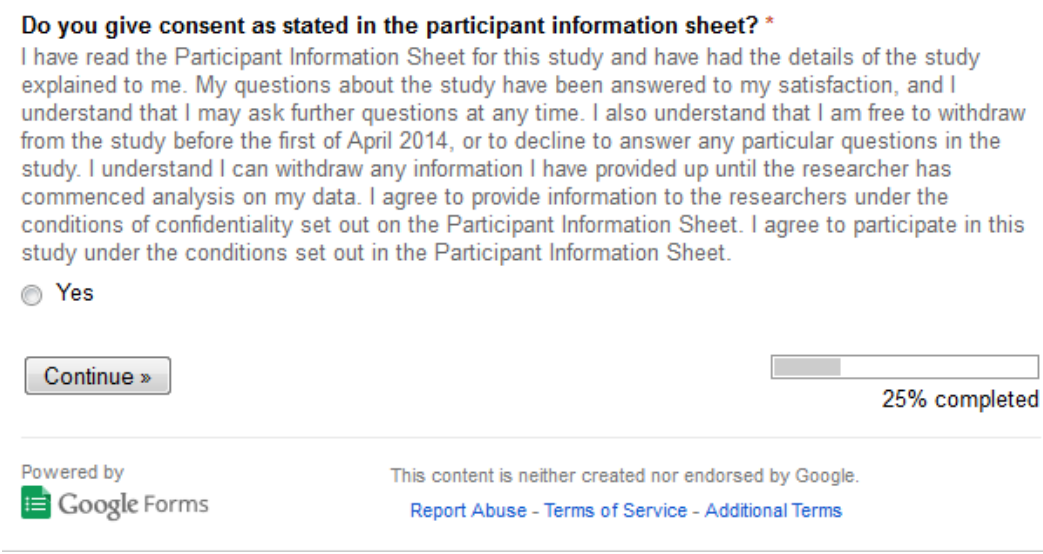
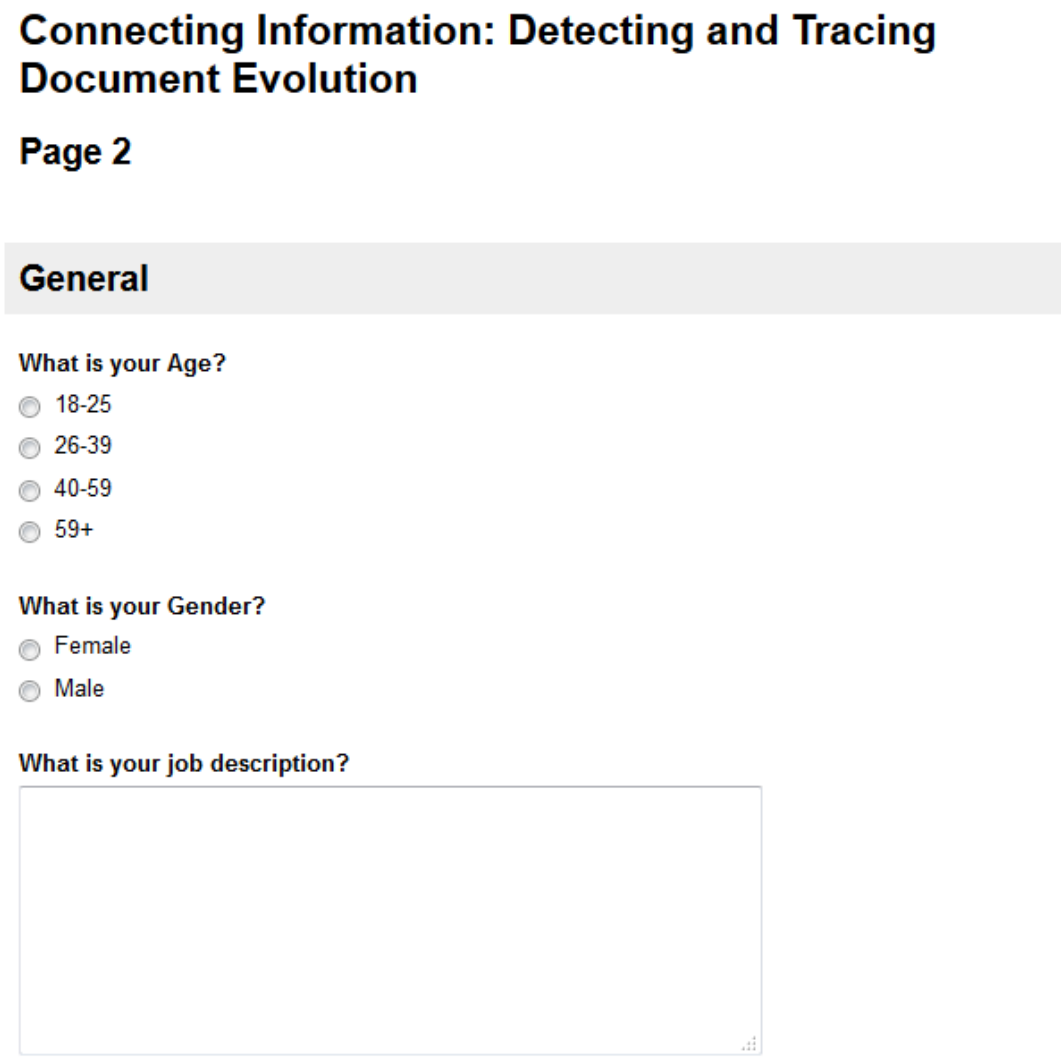


Figure A.7: Study 2 Participant Information — Second Page



Connecting Information: Detecting and Tracing Document Evolution

Page 2

General

What is your Age?

- 18-25
- 26-39
- 40-59
- 59+

What is your Gender?

- Female
- Male

What is your job description?

Figure A.8: Study 2 Questionnaire — First Page Part 1

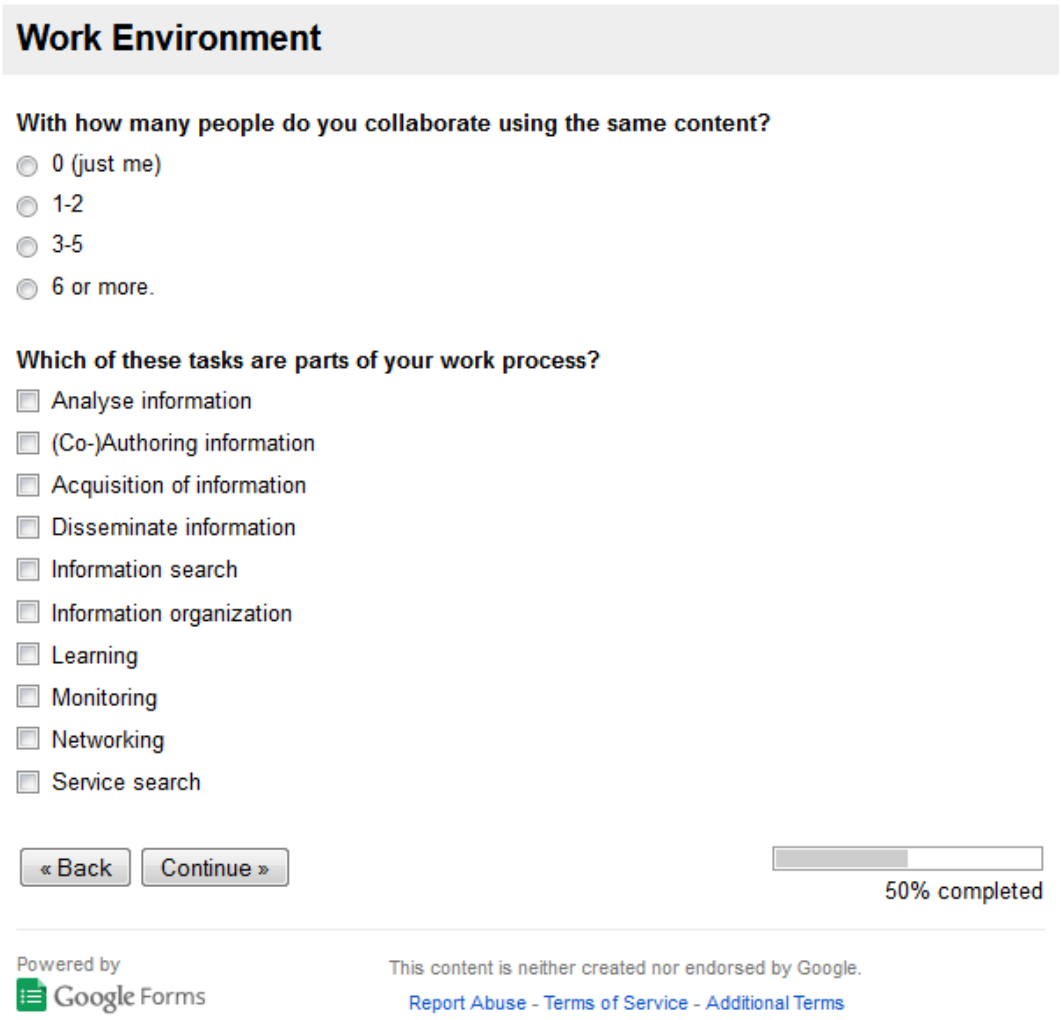


Figure A.9: Study 2 Questionnaire — First Page Part 2

Connecting Information: Detecting and Tracing Document Evolution

Page 3

Software Used

Which content editor do you use most for text documents?

- Ms Word
- Open Office
- Other

If you chose other in the previous question, please specify.

Do you use a document management system?

- No
- Share Point
- Open Text
- Other

If you chose other in the previous question, please specify.

Do you use a versioning system?

- Yes.
- No.
- Do not know.

If no, why not?

- Versioning is not supported by the software I use.
- I do not want versioning.

Figure A.10: Study 2 Questionnaire — Second Page Part 1

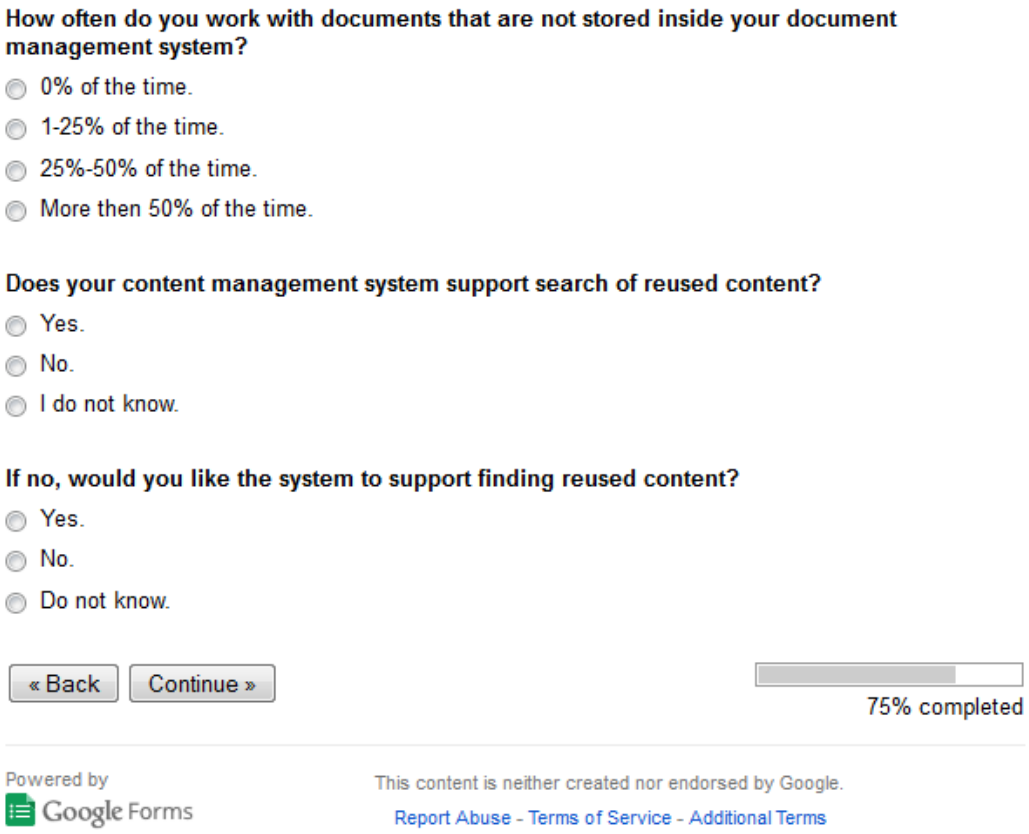
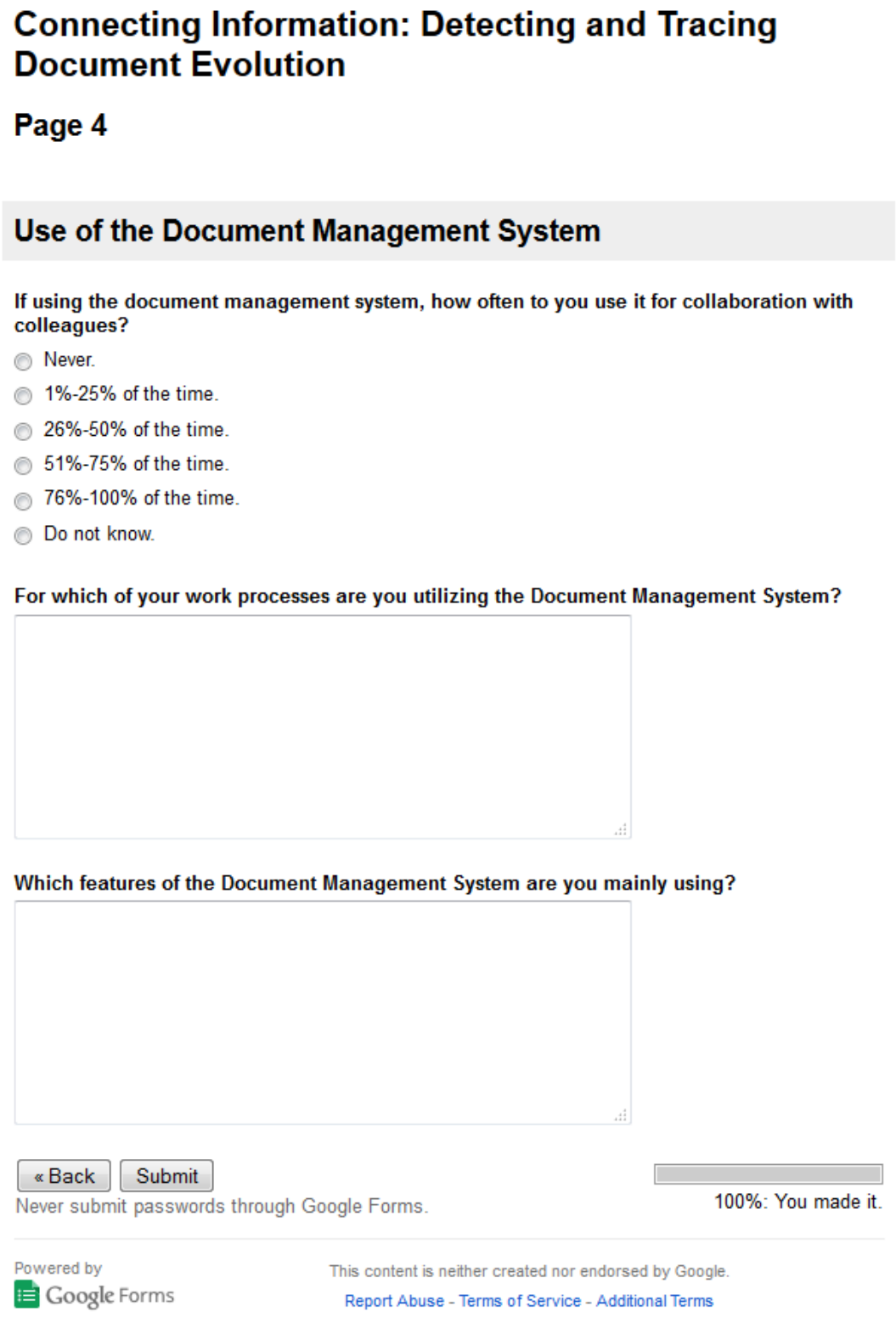


Figure A.11: Study 2 Questionnaire — Second Page Part 2



Connecting Information: Detecting and Tracing Document Evolution

Page 4

Use of the Document Management System

If using the document management system, how often to you use it for collaboration with colleagues?

- Never.
- 1%-25% of the time.
- 26%-50% of the time.
- 51%-75% of the time.
- 76%-100% of the time.
- Do not know.

For which of your work processes are you utilizing the Document Management System?

Which features of the Document Management System are you mainly using?

« Back Submit

Never submit passwords through Google Forms.

100%: You made it.

Powered by Google Forms

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Figure A.12: Study 2 Questionnaire — Third Page

A.3 Exploratory Study — Case Study

This section contains all related material for the third exploratory study. The ethics approval letter is shown in Figure A.13. The participant information sheet for the third study is shown in Figures A.14 & A.15. The interview guideline for the third study is shown in Figure A.16.

Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton
New Zealand

Phone +64 7 838 4021
www.scms.waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

3 March 2014

Michael Rinck
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Michael

Request for approval to conduct a research evaluation involving human participants

I have considered your request to carry out a study for your PhD research project *Connecting Information: Detecting and Tracing Document Evolution* to be conducted by interview and audio recording with staff at Livestock Improvement Corporation, at their request.

The interview will consist of several questions about previous the attempts to install a suitable software system which had failed. The knowledge gained will be used to create a record of the events and analyse the issues that arose.

Questionnaires will not request names of participants therefore any resulting articles and presentations will not contain any personal data and their identities will be kept anonymous. Collected information will be kept in note form or on a password protected computer. Once analyzed and summarized the original notes will be stored in the FCMS data archive till the end of 2018 then destroyed.

The procedure described in your request is acceptable.

The research participants' information sheet, consent form and sample questionnaire meet the requirements of the University's human research ethics policies and procedures.

I therefore approve your application to perform the evaluation.

Yours sincerely,

Masood Masoodian
Human Research Ethics Committee
School of Computing and Mathematical Sciences

Figure A.13: Case Study Ethics Approval Letter

Participant Information Sheet

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Ethics Committee, Faculty of Computing and Mathematical Sciences

Project Title
Connecting Information: Detecting and Tracing Document Evolution.- Case Study

Purpose
This research is conducted as partial requirement for the Ph.D. thesis of Michael Rinck.

What is this research project about?
This project aims to detect and trace the evolution of content. This means that we want to capture the process of content editing and attach the gained information directly to the edited content. This information can then be used to detect the path content took before being used in the document at hand. Users can use this information to discover documents with similar content, in other words related documents.
We would like to do a group interview with a group of researchers from LiC (Livestock Improvement Corporation) on the matter of the introduction of a document management systems. Several attempts have been made before, with no success. We would like to create a record of the events that have taken place when trying to introduce those systems and analyse the issues that arose. The knowledge gained from this will be used in discussing document management systems in general and in relation to our research.

What will you have to do and how long will it take?
You will be asked to participate in an interview at your workplace. You might be asked to show the researcher examples of the techniques you use to store data and keep track of content, e.g. document versions. The researchers will enquire about each of the document management systems you tried to introduce to your work place and explore why they were found not suitable for your work environment. We plan approximately 15-30 minutes for each interview.

What will happen to the information collected?
The information collected will be used by the researcher to substantiate the need to keep track of information about content evolution. It is possible that articles and presentations may be the outcome of the research. The researcher will keep notes and an audio recording of the interviews. Only the researcher and supervisor Annika Hinze will be privy to the notes and recording. All collected data will be kept in note form, locked in a drawer in G.206 or stored on a password protected PC there (audio files). Once analyzed and summarized the original notes and data will be stored in the FCMS data archive till the end of 2018 and then the notes will be destroyed and the audio data will be deleted. No participants will be named in the publications and every effort will be made to disguise their identity, particularly through the use of ID numbers instead of names.

Declaration to participants
If you take part in the study, you have the right to:

- Refuse to answer any particular question, and to withdraw from the study before the first of August 2014
- Ask any further questions about the study that occurs to you during your participation.
- Be given access to a summary of findings from the study when it is concluded.

Who's responsible?
If you have any questions or concerns about the project, either now or in the future, please feel free to contact either:

Researcher:
Michael Rinck
Mr97@students.waikato.ac.nz

Supervisor:
Annika Hinze
hinze@cs.waikato.ac.nz

Figure A.14: Study 3 Participant Information — First Page

Research Consent Form

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Ethics Committee, Faculty of Computing and Mathematical Sciences

Connecting Information: Detecting and Tracing Document Evolution – Case Study

Consent Form for Participants

I have read the **Participant Information Sheet** for this study and have had the details of the study explained to me. My questions about the study have been answered to my satisfaction, and I understand that I may ask further questions at any time.

I also understand that I am free to withdraw from the study before the first of August 2014, or to decline to answer any particular questions in the study. I understand I can withdraw any information I have provided up until the researcher has commenced analysis on my data. I agree to provide information to the researchers under the conditions of confidentiality set out on the **Participant Information Sheet**.

I agree to participate in this study under the conditions set out in the **Participant Information Sheet**.

Signed: _____

Name: _____

Date: _____

I agree / do not agree to my responses to be audio recorded.

Signed: _____

Name: _____

Date: _____

Researcher's Name and contact information: Michael Rinck
mr97@students.waikato.ac.nz

Supervisor's Name and contact information: Annika Hinze
hinze@cs.waikato.ac.nz

Figure A.15: Study 3 Participant Information — Second Page

Particpant ID: Age: Gender: Role:

System	Like	Dislike	Likert Rating
Deki Wiki			
Open Atrium			
Alfresco			
Word press			
Media Wiki			

Figure A.16: Study 3 — Interview Guideline

A.4 Data Acquisition

This section contains all related material for the data acquisition. The ethics approval letter is shown in Figure A.17. The participant information sheet for the data acquisition is shown in Figures A.18 & A.19.

Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton
New Zealand

Phone +64 7 838 4021
www.fcms.waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

14 February 2012

COPY

Michael Rinck
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Michael

Request for approval to conduct a research evaluation involving human participants

I have considered your request to carry out a study for your PhD research project *Connecting Information: Detecting and Tracing Document Evolution* to be conducted with University of Waikato staff members.

You will be conducting a diary study with users whilst tracking the content flow on their work machines with the use of a Microsoft Word Addin, capturing the process of content editing, asking the participants to use that data and finally, conducting an interview about the experience.

I note that the software you will use will be given to TSG for testing for safety before the experiment commences. You will ensure that all data on participants' machines is backed up before installing and de-installing any software. No actual content will be accessible to the researcher, only how and when content was edited and moved.

Data will be collected anonymously using a random identity number.

Publications and reports will not contain any personal data of the participants and their identities will be kept anonymous. Collected information will kept in note form or on a password protected computer. Once analyzed and summarized the original notes will be destroyed and all digital notes and data will be deleted.

The procedure described in your request is acceptable.

The research participants' information sheet, consent form and diary sheet meet the requirements of the University's human research ethics policies and procedures.

I therefore approve your application to perform the evaluation.

Yours sincerely,

LA Hunt.

Lyn Hunt
Human Research Ethics Committee
Faculty of Computing and Mathematical Sciences

Figure A.17: Data Acquisition Approval Letter

Participant Information Sheet

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Ethics Committee, Faculty of Computing and Mathematical Sciences

Project Title
Connecting Information: Detecting and Tracing Document Evolution.

Purpose
This research is conducted as partial requirement for the Ph.D. thesis of Michael Rinck.

What is this research project about?
This project aims to detect and trace the evolution of content (text in word files). This means that we want to capture the process of content editing and attach the gained information directly to the edited content (the word files). This information can then be used to detect the path content took before being used in the document at hand. Users can use this information to discover documents with similar content, in other words related documents.

What will you have to do and how long will it take?
You will be asked to participate in a diary study for 4 weeks, noting down any problems related to digital content as they happen. We do not expect the diary entries to take up more than 10 minutes a day. You will also be asked if we may install software on your work machine that tracks the content flow in between documents. The software is a Microsoft Word AddIn. No actual content is accessible to the researcher at any time, only the how and when content was edited and moved. We will analyze how much data was accumulated in this month, e.g. how many transactions were recorded and how many documents were involved.
After this period, we would like to install an interface to the software on your work machine, which you may use for another month. This interface will allow you to access the data collected and use it to improve your workflow. At the end of the second month, we would like to interview you about the experience. We will then de install all software used. The software used was tested by ITS (Information & Technology Services Division) for safety before the experiment starts. This means ITS will check if the software will cause problems with your current set up of your work machine. We will require all data on the used pc to be backed up before the software is installed and before it is deinstalled.

What will happen to the information collected?
The information collected will be used by the researcher to substantiate the need to keep track of information about content evolution. It is possible that articles and presentations may be the outcome of the research. The researcher will keep notes of the interviews and diary study notes. Only the researcher and supervisor Annika Hinze will be privy to the notes. All collected data will be kept in note form, locked in a drawer in G.206 or stored on a password protected PC there. Once analyzed and summarized the original notes and data will be stored in the FCMS data archive till the end of 2018 and then be destroyed. No participants will be named in the publications and every effort will be made to disguise their identity, particularly through the use of ID numbers instead of names.

Declaration to participants
If you take part in the study, you have the right to:

- Refuse to answer any particular question, and to withdraw from the study before the first of December 2013
- Ask any further questions about the study that occurs to you during your participation.
- Be given access to a summary of findings from the study when it is concluded.

Who's responsible?
If you have any questions or concerns about the project, either now or in the future, please feel free to contact either:
Researcher:
Michael Rinck
Mr97@students.waikato.ac.nz
Supervisor:
Annika Hinze
hinze@cs.waikato.ac.nz

Figure A.18: Data Acquisition Participant Information — First Page

Research Consent Form

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Ethics Committee, Faculty of Computing and Mathematical Sciences

Connecting Information: Detecting and Tracing Document Evolution

Consent Form for Participants

I have read the **Participant Information Sheet** for this study and have had the details of the study explained to me. My questions about the study have been answered to my satisfaction, and I understand that I may ask further questions at any time.

I also understand that I am free to withdraw from the study before the first of December 2013, or to decline to answer any particular questions in the study. I understand I can withdraw any information I have provided up until the researcher has commenced analysis on my data. I agree to provide information to the researchers under the conditions of confidentiality set out on the **Participant Information Sheet**.

I agree to participate in this study under the conditions set out in the **Participant Information Sheet**.

Signed: _____

Name: _____

Date: _____

Researcher's Name and contact information: Michael Rinck
mr97@students.waikato.ac.nz
Phone: 0220783017

Supervisor's Name and contact information: Annika Hinze
hinze@cs.waikato.ac.nz
Phone: Extension 4052

Figure A.19: Data Acquisition Participant Information — Second Page

A.5 User Study — Evaluation

This section contains all related material for the evaluation study. The ethics approval letter is shown in Figure A.20. The participant information sheet for the evaluation study is shown in Figures A.21 & A.22. The task sheet for the evaluation study is shown in Figures A.23 & A.24.

Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton
New Zealand

Phone +64 7 838 4021
www.scms.waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

26 March 2014

Michael Rinck
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Michael

Request for approval to conduct a research evaluation involving human participants

I have considered your request to carry out a study for your PhD research project *Connecting Information: Detecting and Tracing Document Evolution (in lab tests)* to be conducted in the Usability Lab, Computer Science Department.

The goal is to test the prototype which was developed to track and visualize how content was copy, pasted and edited between documents in MS word files. The results will be in your PhD thesis.

We note that publications and reports will only include anonymized data and that ID numbers will be used if specific details are named in reports, which participants will have given consent.

The procedure described in your request is acceptable.

The research participants' information sheet, consent form and sample questionnaire meet the requirements of the University's human research ethics policies and procedures.

I therefore approve your application to perform the evaluation.

Yours sincerely,

Masood Masoodian
Human Research Ethics Committee
School of Computing and Mathematical Sciences

Figure A.20: Evaluation Study Ethics Approval Letter

Participant Information Sheet

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Ethics Committee, Faculty of Computing and Mathematical Sciences

Project Title
Connecting Information: Detecting and Tracing Document Evolution.- In Lab Tests

Purpose
This research is conducted as partial requirement for the Ph.D. thesis of Michael Rinck.

What is this research project about?
This project aims to detect and trace the evolution of content. This means that we want to capture the process of content editing and attach the gained information directly to the edited content. This information can then be used to detect the path content took before being used in the document at hand. Users can use this information to discover documents with similar content, in other words related documents. A prototype was developed to track and visualize how content was copy pasted and edited between documents. This study is used to assess how useful the prototype is for solving tasks related to document management.

What will you have to do and how long will it take?
You will be asked to participate in an experiment hosted at the Department of Computer Science. You will be asked to execute several tasks organizing documents on a computer. You will be assigned to Group A or B. Group A will execute the given tasks. Group B will get to use prototypic software to support executing the tasks. The researcher will explain how the prototypic software is used. The researcher will stop the time needed to execute these tasks and analyze the accuracy of the results of the tasks. After the experiment, the researcher will ask a few questions about the experience. The overall time for introduction, experiment and questions will be no longer than 30 minutes.

What will happen to the information collected?
The information collected will be used by the researcher to analyze the usefulness of the prototypic software. It is possible that articles and presentations may be the outcome of the research. The researcher will keep notes of the experiment. Only the researcher and supervisor Annika Hinze will be privy to the notes. All collected data will be kept in note form, locked in a drawer in G.206 or stored on a password protected PC there. Once analyzed and summarized the original notes and data will be stored in the FCMS data archive till the end of 2018 and then the notes will be destroyed and the data will be deleted. No participants will be named in the publications and every effort will be made to disguise their identity, particularly through the use of ID numbers instead of names.

Declaration to participants
If you take part in the study, you have the right to:

- Refuse to answer any particular question, and to withdraw from the study before the first of October 2014
- Ask any further questions about the study that occurs to you during your participation.
- Be given access to a summary of findings from the study when it is concluded.

Who's responsible?
If you have any questions or concerns about the project, either now or in the future, please feel free to contact either:

Researcher:
Michael Rinck
Mr97@students.waikato.ac.nz

Supervisor:
Annika Hinze
hinze@cs.waikato.ac.nz

Figure A.21: Study 4 Participant Information — First Page

Research Consent Form

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Ethics Committee, Faculty of Computing and Mathematical Sciences

Connecting Information: Detecting and Tracing Document Evolution – Case Study

Consent Form for Participants

I have read the **Participant Information Sheet** for this study and have had the details of the study explained to me. My questions about the study have been answered to my satisfaction, and I understand that I may ask further questions at any time.

I also understand that I am free to withdraw from the study before the first of October 2014, or to decline to answer any particular questions in the study. I understand I can withdraw any information I have provided up until the researcher has commenced analysis on my data. I agree to provide information to the researchers under the conditions of confidentiality set out on the **Participant Information Sheet**.

I agree to participate in this study under the conditions set out in the **Participant Information Sheet**.

Signed: _____

Name: _____

Date: _____

Researcher's Name and contact information: Michael Rinck
mr97@students.waikato.ac.nz

Supervisor's Name and contact information: Annika Hinze
hinze@cs.waikato.ac.nz

Figure A.22: Study 4 Participant Information — Second Page

Participant Information

Participant ID:

Age: Gender:

Profession:

Task 1:

The document c:\Task1\January 2014\Result letter Bela Lugosi.docx was created with the help of template(s) from the folder c:\Task1\Discipline letter templates. This means that content was copy pasted from the templates to the letters, or the template edited into a letter and saved as the letter.

Which templates were used?

What other files in the folder c:\Task1\January 2014\ were created with the use of these templates?

How confident are you that your results are correct and complete on a scale from 1 (no confidence) to 5 (full confidence)?

Figure A.23: Study 4 — Questionnaire Page 1

Task 2:

The folder c:\Task2\ includes all confirmation letters and templates for staff appointed to undertake Summary Jurisdiction hearings. Please list staff that was re-appointed and staff that was newly appointed for the role and when.

How confident are you that your results are correct and complete on a scale from 1 (no confidence) to 5 (full confidence)?

Figure A.24: Study 4 — Questionnaire Page 2