View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Research Commons@Waikato

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Research Commons

http://researchcommons.waikato.ac.nz/

Research Commons at the University of Waikato

Copyright Statement:

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

The thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author’s right to be identified as the author of the thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author’s permission before publishing any material from the thesis.

Evolution of documents:
Information and Data objects

A thesis

Submitted in fulfilment

Of the requirements for the degree

Of

Master of Science in Computer Science

At

The University of Waikato

By

APPU MATHEW JOSE

KO TE TANGATA

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

The University Of Waikato

2012

This page is intentionally left blank

Abstract

The vast amount of data leading to the digital data explosion highlights the fact that the current data definition needs a change as the current systems are unable to track the evolution in a document over time without manual intervention. The concepts of Information and Data objects are introduced in this thesis to track the evolution information in a document. We developed the requirements for such a system in which the evolution information is automatically tracked by tracking the user copy and paste action and then using the data to create the evolution information about a specified document. A case study is discussed to further analyse the information and data flow in a collaboration. We have used this knowledge to design the system and then to implement the system so that the user copy and paste actions can be tracked to create the evolution information. The implementation is then presented to a group of experts to identify the problems and to get the feedback to improve the system.

Table of Contents

List of Figures

1 Introduction

The introduction of surface computing and gesture recognition has taken the computing to the next level, but document-related concepts, however, remain largely unchanged. There is a need for a new way of representing data at the document level and the concepts of document needs a change from the olden days. Rinck and Hinze [RH11] discuss the need for a new document definition, a change to the existing document concepts, based on the user study they have conducted.

We feel that the idea of considering the document as a single entity needs to be changed. With the existing document concepts, it is impossible to track the evolution of changes that happened to a document when you consider the document in a collaborative scenario. Although the versioning systems[1] have evolved to support the collaborative scenario, they mainly track the evolution manually, that is, users will have to manually enter the comments into the versioning systems. The time machine on Mac OS is a backup utility and does not track the relationships between the documents. This thesis will address this problem of tracking the changes in a collaborative environment without any manual intervention by suggesting a new concept of representing documents.

We will first discuss a scenario in a collaborative environment as to how information can be shared and how users collaborate. We have illustrated this scenario so that we can further study about the current document concepts and its limitations in a collaborative environment. This scenario will illustrate how users collaborate and the difficulties they face during such collaboration under the current document concepts. We will use this scenario to further explore what this thesis would contribute.

[1] Such as cvs,tfs

1.1 Scenario

In this section we present a scenario between three scientists during a conference about information sharing and collaboration

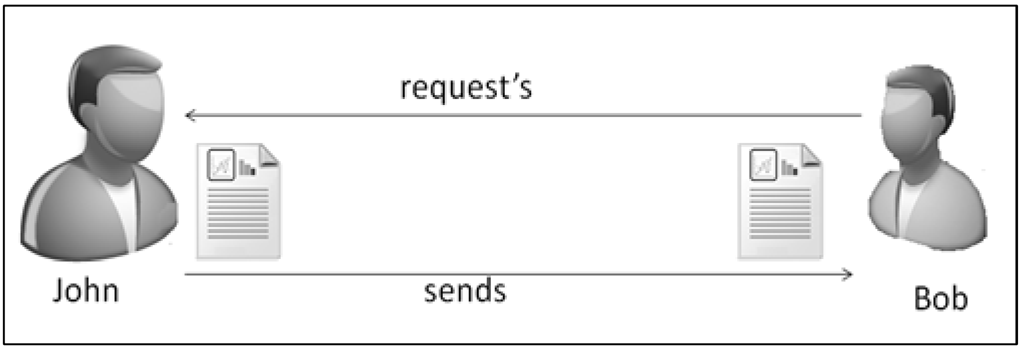


Figure 1.1: Scenario representation

Bob and John met during a scientific conference and they were soon discussing about their works in the field of advanced computing. Bob became very interested in some part of John's work. Bob felt that a part of the work done by John could actually propel his new paper. So he consults John and asks his permission to use his comparative graph and findings in his new paper. John readily agrees and provides with all the information.

Months after they met, John felt intrigued about part of his study given to Bob and he also wanted to know how far Bob has completed his paper and the extent to which his contributions helped. Bob had made some changes to John's graph and use some of John's material. However, Bob found it very difficult to point out exactly what changes he had made at what stage of writing the paper.

When we take a closer look at the scenario, we can see that this is not possible until and unless there is some other ways of representing data other than containing all the data into a single document. The new ways of data representation is important as the existing methods cannot be modified or rather they cannot be modified to suit our needs. Each file type is having prescribed or standard metadata information attached to it. Meta data primarily means data about the data. Even the metadata information is necessarily unable to track down the evolution path of data.

1.2 Contribution of this thesis

In this section we point out what this thesis contributes to current research. The thesis covers the following aspects:

- ✓ Study the current document concepts
- ✓ Evaluate the problems of current document concepts
- ✓ Propose a new document concept (Information objects and Data objects)
- ✓ Explore the new concept with a case study
- ✓ Implement a proof of concept application

1.3 Thesis structure

This section describes how the thesis is structured. The remainder of the thesis is divided into following seven sections:

Chapter 2: In Chapter 2 we will introduce the concepts of Information objects and Data objects and will define them. We will also have a look at the possible systems viz., MS Word and Latex and will come up with the requirements to be met by the proposed system.

Chapter 3: Chapter 3 will be related works where we look into all possible related works in this area. We will look into each study and evaluate with a set of criteria. This will enable us to see whether the existing study is specific on our set of criteria. We will conclude this chapter by comparing each study to the set of criteria of our proposed system. This set of aspects would be derived from the chapter 2 where we discuss about the concepts of Information objects and Data objects. These aspects would be core criteria of the proposed system.

Chapter 4: In this chapter, we look into the information and data flow by introducing a case study which is a scenario that has happened and involves collaboration. We will describe the scenario as described by the persons involved. We will look deeply into this case study and will study about how evolution progresses while collaborating and to refine the concept of information objects based on this study. This chapter will finally analyse our concepts based on the case study.

Chapter 5: In Chapter 5 we will look into the design of the proposed system. We will sub-divide the system into various components. We will also put forth a detailed architectural diagram to detail the working of the system. We will also describe how the system would work for the case study in chapter 4 to further make the users understand about the concepts of information object and Data object.

Chapter 6: In this chapter, we will document, in detail, the first phase of implementation of the project. We will first look into various open source projects that we could modify to meet the requirements of the proposed system. We will then do a comparative study between the projects with those requirements to see the effectiveness of the system. We will explain the system in detail by providing screen shots and then detailing the working of each screen.

We will also discuss about the various challenges that we faced during the implementation phase and detail out how we resorted to those challenges.

Chapter 7: In chapter 7, we will document the expert walkthrough. A software walkthrough was provided to each participating expert and then a group walkthrough to the group of experts. We will detail the outcome of each of the walkthrough. The walkthrough is aimed at getting more feedback about the system which will lead us to identifying the problems in the system and thus help us improve the quality of the system. We will also document all the suggestion which was provided by the experts for improving the quality of the system.

Chapter 8: In Chapter 8, we will conclude this paper along with a discussion to the future works. We will also include new ideas for the future work.

2 Data objects and Information objects

In this chapter, we will define the Data objects and Information objects based on the scenario. The main aim of this chapter is to develop the concept of Information object and Data object to help implement the system. We will also look at possible systems that are widely in use. We will then conclude this chapter by analysing the requirements of the proposed system.

2.1 Scenario revisited: Analysis of scenario

In this section we will analyse the scenario in the previous chapter.

From Fig 1.1 we can easily understand the process that happened in the scenario. Below is a synopsis of what happened:

- Bob became interested in some parts of John's work
- Bob asks John for the parts
- John sends it to Bob
- John wants to know how his parts helped Bob

With the current document concepts, it would be impossible for John to know what has happened to his study without asking Bob. In current document concepts, parts of document are not given significance. Document as a whole is treated a single entity and the parts or data are building blocks. The current document management systems do not track the origin of the parts.

So we feel that there should be a system which is capable of tracking down the origin of parts or rather the evolution of a document from its inception. We also want the system to give significant importance to parts of document by tracking the source so that repetition of data can be significantly avoided.

If we had a system like this John would have definitely able to track down what happened to his parts of document and whether it had actually helped Bob.

Further more from a researcher's point it is all information that adds up the knowledge.

2.2 Concepts

The following are the concepts that we would like to discuss in this chapter.

- ❖ Data object: Data are raw facts and so Data object is a collection of raw variables. For example, a graph, a picture or a text block can be considered as a Data object.

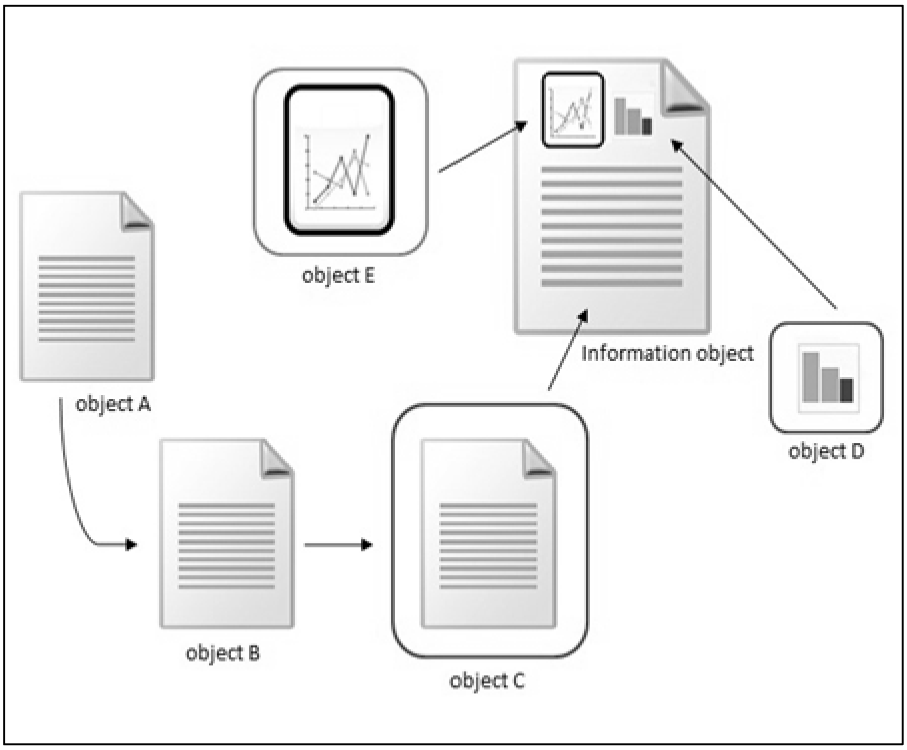


Figure 2.1: Evolution of Bob's paper as Information object evolution

- ❖ Information objects: Information objects are a meaningful collection of Data objects in a particular order. So Information objects can be defined as a logical collection of Data objects.

The Figure 2.1 is a diagram showing the Information object and Data object concept and gives a conceptual view of how data is represented and how the evolution is tracked if we were to use the concept of Information objects. From the Figure 2.1 Information object is a collection of Data objects C, D and E. The Data object C is evolved from the Data objects A and B. The Data object C represents John's findings and object E represents the graph he used.

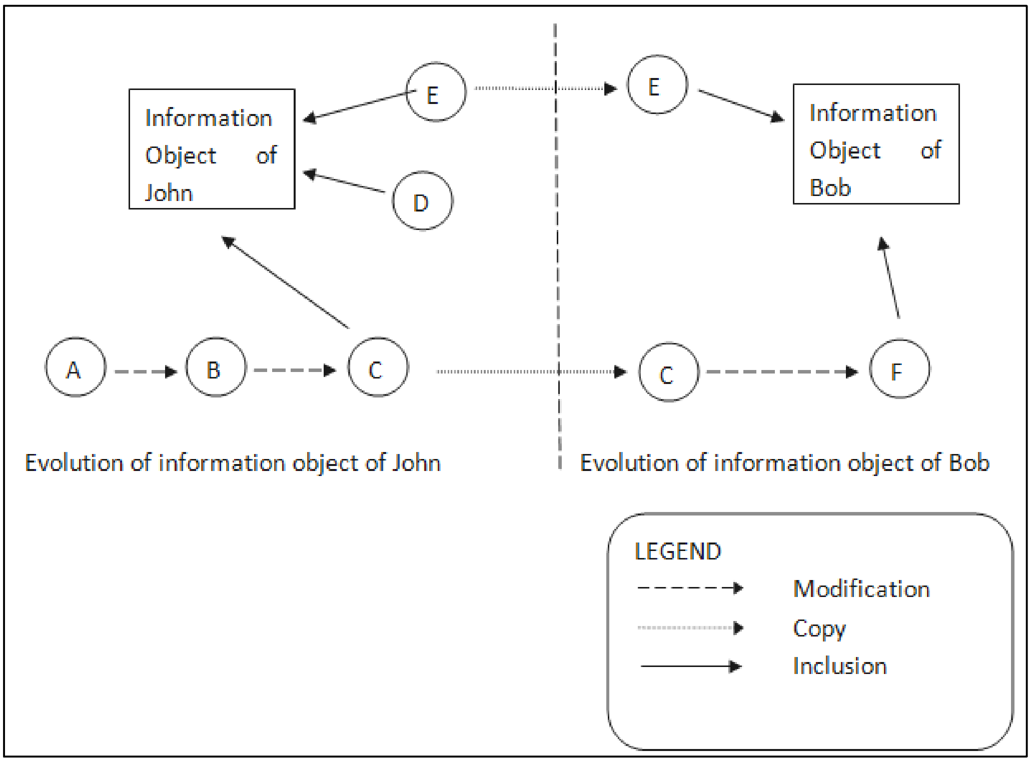

Information Object of John
E
D
A
B
C
Evolution of information object of John
E
Information Object of Bob
C
F
Evolution of information object of Bob
LEGEND
Modification
Copy
Inclusion

Figure 2.2: Evolution of Information object based on scenario

When John gave the findings and the comparative graph to bob, it was actually the Data objects C and E that got transferred to Bob (see Figure 2.2). Bob used the comparative chart (object E) as such in his paper and then went on modifying the object C with some more additions to the findings part which transforms the Data object to F. So it's the objects E and F that Bob uses in his study.

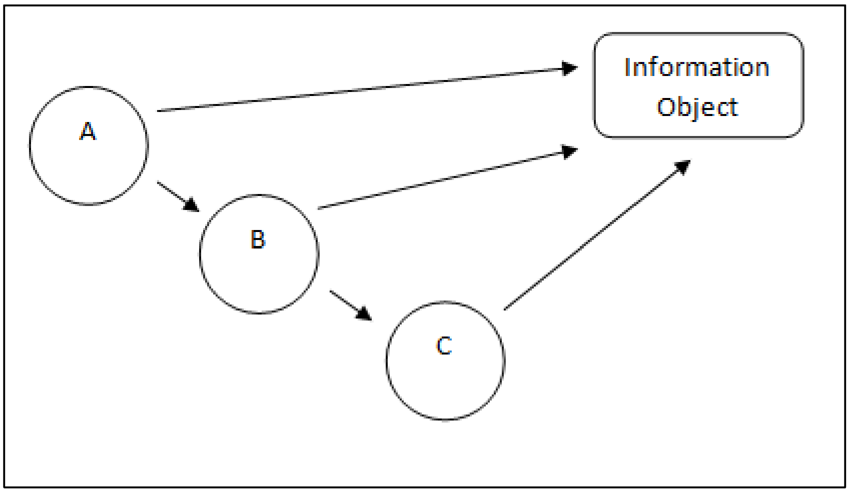


Figure 2.3: Data objects and Information objects

We will now define and differentiate between a Data object and an Information object. A Data object (see Figure 2.3: A, B and C denote version of the Data object) is the basic building block of an Information object and is independent of the Information object in which it is used. An Information object is the collection of different types of Data objects. The data objects convey meaningful information when they are combined. For example, an Information object is more or less same as the document in the present systems, which are typically collections of texts and pictures. Each text and picture will be a Data object in the given context. The evolution tree is the representation of the Data object that is evolving (i.e., being changed over time).

Comparison to the Transclusion Principle

Ted Nelson defines Transclusion as the *re-use of the original content through embedded shared instances* [NEL95]. According to him, transclusion brings to electronic publishing a copyright method that makes republication fair and clean [NEL95]. From the figure 2.4 it can be easily understood that the prime focus of transclusion is the re-use of the original content thus giving the principle a unique ability to implement it as a copyright method. Also the transclusion principle was developed mainly as a document management system.

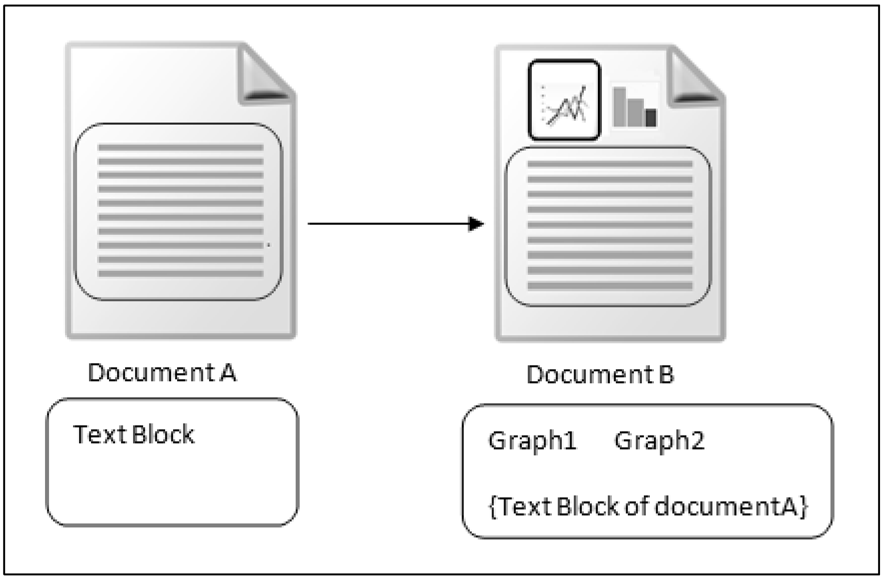


Figure 2.4: Example of a simple transclusion

From the concepts stated in section 2.2, it could be mistaken that the data and information object also points to the direction of Transclusion principle. This is not true. Both the concepts treats data as the root object as against the common practice of treating a file as the root object. This thesis mainly focuses on the evolution of the document rather than focusing on the re-use of the original content. It is also true that the original content can be re-used by effectively tracking the evolution but evolution cannot be tracked by tracking the re-use of original content.

2.3 A closer look at current systems

In this section we will take a closer look into the word processing systems. We try to differentiate the difference in the working of MS Word and Latex, which are both word processing systems.

2.3.1 MS Word

MS Word [MSWD] has import features in which you can insert a picture or just copy paste a picture. When using the picture the image object is brought inside the word processor and hence is no way references the original file. The imported image object is then a part of the word processor and it facilitates the auto-formatting of the imported image object.

It also has another import feature which is to import objects or objects from file belonging to different file types. Even though the object is linked from the file no reference is maintained and hence any changes to the original object or file will not be reflected in the document.

Another feature in the MS Word is the outlining feature which allows us to create a master and sub-document. When using this feature we can have a master document which outlines the subdocument and then go on with making changes to the subdocuments. For example, if we have a large book to be written in word, we can have a master document which links to all the chapters in that book. Any changes made to be subdocuments will be reflected in the main document. The limitation as we see it is that, only word format documents can be linked or inserted in the outlining. No other file formats or independent Data objects can be inserted.

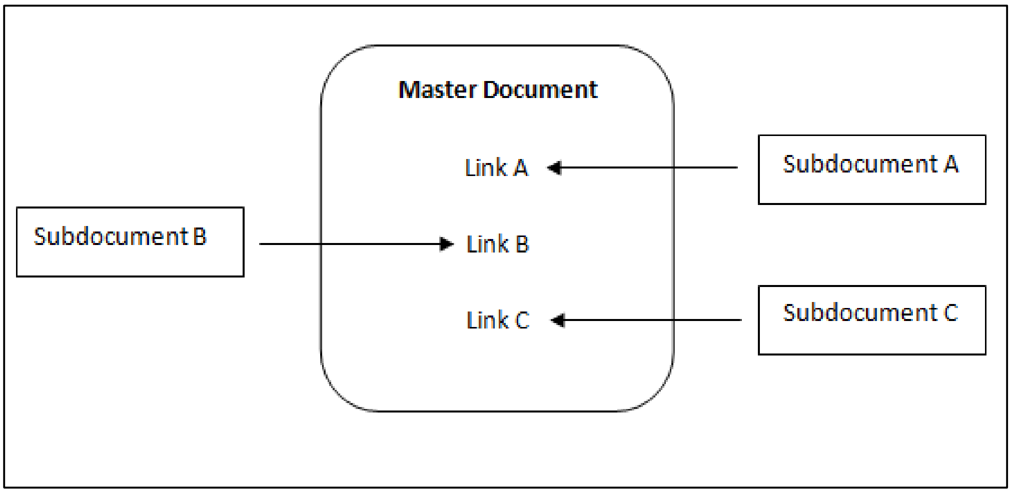


Figure 2.5: Outlining in MS Word

2.3.2 Latex

Latex [LTX] is a document preparing system in which the latex type-setting platform is used to develop documents. The documents are most commonly in pdf format (Portable Document Format). When the document is produced or the pdf is made from the latex development platform, the resulting pdf document will have objects embedded to the document and will not maintain any reference. So the resulting pdf document conforms to the standard of the PDF's and maintains no link as a reference. The following code shows how a picture 'image.jpg' can be included to the Latex document.

```
\usepackage[pdftex]{graphicx}
\begin{document}

Text

\includegraphics{image.jpg}

Text
\end{document}
```

The type-setting platform however maintains only references. For example, to type-set an image the latex platform should maintain the reference to the external file. Unlike the ms-word, the latex type-set platform does not import the object and provide formatting options. So the principle of Transclusion [NELS95], *the re-use of the original content through embedded shared instances*, are partly supported in the type-set platform.

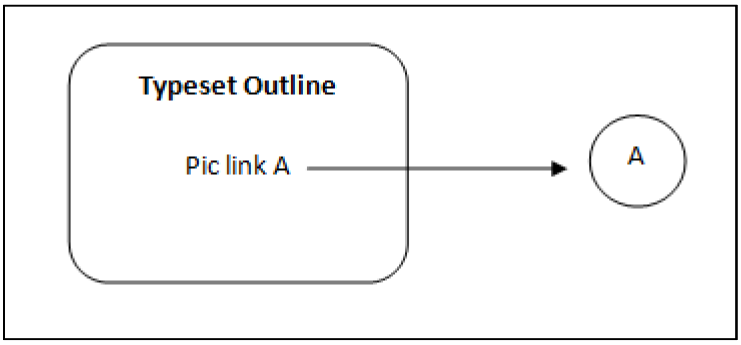


Figure 2.6: Outlining in Latex

We also feel that it is good to discuss about master and child relationship in the web since we are comparing possible transclusion in the real world. The asp.net provides the concept of the master pages when it comes to web-development. A single master page defines the look and feel and standard behaviour that we need for all the web pages (or a group of pages) in the application. We then create an individual content page that contains the content that the website owner wants to display. When users request the content pages, the individual content pages merge with the master page to produce output that combines the layout of the master page with the content from the content page [MP]. Since it is web based, the support is limited to HTML only and not any other file types.

So it's imperative that none of the systems that are available today implements the transclusion principle or the concept of the independent Data objects. The major drawback is that every system is developed with a view to attain perfection within its own purview i.e. the word supports only the outlining of only word documents and no other file types. Also the MS-Word does not consider the idea of having independent Data objects. It considers the document as whole entity.

2.4 Requirements

To realize the concept of Information objects and independent Data objects, the following requirements must be satisfied.

- **R1**: A different way of representing data as object – The existing meta-data representation of the various file types is not sufficient to include the evolution information into the metadata and so a different way of representing data is inevitable.
- **R2**: The Data object should be independent of the Information object in which it is used
- **R3**: The Data Object should be able to contain the evolution information in itself – For example in the figure 2.2; the Data object C should know

the predecessor A and B. Each Data object shall contain information to its immediate node. For example, the Data object B will contain information about object A (source) and object C (destination). Similarly, the Data object C will contain information about object B (source) only and not about object A.

- **R4**: The evolution information should get embedded into the Data object without any manual intervention – whenever there is a change happening like a copy-paste or drag and drop by which there is any change to the Data object the change information or the evolution information should be automatic and should happen without any manual intervention.

2.5 Summary

From this chapter, we were able to define the Information object and Data object. We were also able to analyse the scenario to stress the importance of a new system and that the current document concepts need a change. We were also able to refine those concepts by doing a comparative study between MS word, a word processing application and Latex, a type set platform. We were also able to derive the set of requirements for the proposed system through the comparative study.

3 Related Work

In this chapter, we look into all possible related work that can be considered in the document evolution context along with versioning system, transclusion principle; hyper texting and other inter connected studies.

The evaluation of the study is based on the following four aspects:

- **Hierarchy of parts** –The hierarchy refers to arrangement of items or data in the document. When the data is actually represented each part of the data will have the position attached to it, referred to as the hierarchy of parts. The Hierarchy of parts is important as our proposed system consider the data as a single entity which is independent from the document. From the Figure 2.1 we can easily understand that hierarchy of data and how it is arranged is very important. This is because of the fact that information is meaningful when the data is arranged in a logical order or hierarchy. So the hierarchy of parts is a significant aspect as the Information object can never be meaningful when the Data objects are scattered.

- **Evolution** – refers to the ability to track the changes and the gradual changes that is happening to the document as a whole or in part. The evolution in our proposed system is the ability of the independent Data object to carry with in itself the evolution information as to how it has evolved over time. When referring to the scenario in figure 1.1 we can analyse that tracking evolution of document is one of the important aspect and reason why needed a new system and document concept instead of the existing document concept where the parts of document is not tracked for origin.

- **Granularity** – refers to the level of sub-division as far as the data is considered. When considering a document, the granularity refers to the level in which the document and its data are sub-divided which helps a great deal in collaboration. The granularity of Data objects essentially is the amount to which an independent Data object can be broke down into so that the concept of Data object in the proposed system can be streamlined. Taking a look at the scenario and Figure 1.1 John could have easily given his comparative graph and its findings if the data was independent. To enhance collaboration we need granular data which means data which is independent of others.

- **Including parts of document** – refers the capability of the document to keep the linking to another document when the data or part of data is being referred. The proposed system brings forth the idea of considering the data as independent objects and hence objects of data will be linked to the document rather than making the Data objects a part of the document itself. From figure 2.1 the evolution of data can be tracked for changes only if the objects are independent. When the objects are independent we could easily link to the objects from the document. By linking we do not mean importing the object into the document but referencing the object so that we can track the origin thereby helping to form the evolution tree.

3.1 Related Study

Now we will look into the various studies and systems which are already in place and which are of interest to us in building the proposed system.

3.1.1 Software Systems

All the version control systems such as the CVS [CVS], SVN [SVN] and TFS [TFS] are software which keeps tracks of the changes that is done to a particular file and helps people to collaborate. Most of them are based on the client-server architecture where the files are actually stored in the central repository. The files will be required to be checked-out either by a single person or multiple people for editing. The TFS builds the hierarchy of changes in file by analysing the edits done to a file. It considers the file as one and does not break up into parts. The versioning systems also support the evolution very effectively if the information regarding each edit and change is given as a comment or a note manually. The versioning system considers data hierarchy and hence it is supported partially. It is able to track the evolution to a particular document and hence it is fully supported.

3.1.2 DEVAC: evolution of documents by Spatio-temporal analysis

Ryu et al. [RKC08] developed a software DEVAC (Document EVolution Analyzing Centre). This is basically plagiarism detection software which is capable of creating a phylogeny tree of the related documents. Even though the plagiarism is not our point of interest, we were interested in the software which was able to track down the evolution of two given documents. They proposed that both a spatial and a temporal analysis of a document is indeed essential to detect plagiarism or even to say whether two documents are related to each other. It does not break down the document into further small parts and considers as a single entity. The DEVAC system supports the evolution tracking fully and a partial support of the hierarchy of parts since it considers the spatial and temporal analysis.

3.1.3 A web based approach to Transclusion Principle

Krottmaier and Maurer [KM01] in their paper put forth some ideas to implement the transclusion principle, originally suggested by Ted Nelson in 1960’s, using a Hyperwave Information Server (HIS).

The transclusion [NELS95] is the re-use of the original content through embedded shared instances. In other words, Transclusion (see fig 2.4) is the inclusion of content of one document to another document through reference.

The paper however is limited to the common format types present in the WWW and which uses the HTTP protocol. According to them, the transclusions principle can be implemented, since the original text is included to another document, a server in the Internet is almost necessary. According to the transclusions principle the reading of a quote in its original context will be of more interest in matters pertaining to research. It fully supports the linking the other part of the document using the WWW and HTTP protocol. It is also able to partially track the evolution as there are links to other files which can be tracked.

3.1.4 Transclusions in HTML Environment

Kolbitsch and Maurer [KM06] proposes another implementation technique in the HTML based environment. They suggest the use of XLink, XML Linking Language, for the implementation of the transclusions in the HTML area. They also suggest the use of IFRAMES and Embedded Objects but does not recommend as it is very limited when it comes to implementation. They recommend the XLinK, even though it is slow, as it is accurate when it comes to actually implementing transclusions. To implement transclusion in any form, even thought restricted to HTML format, they broke down the HTML based tags to have value based on reference to servers where data is held. They support including parts of document by using the WWW and HTTP protocol and partially support the evolution feature. Due to the use of IFRAMES and the embedded objects they also extend a partial support to the granularity.

3.1.5 Comprehensive File Versioning Systems (CVFS)

Soules et al. [SGSG03] proposes the uses of Comprehensive File versioning Systems and the paper examines the use of two state efficient metadata structures for versioning systems. They proposed the log based and the multi-version b-trees for the implementation of the CFVS. The CVFS is able to track down the evolution between any two documents very effectively since the CVFS uses the underlying principle used in the modern versioning systems. The CVFS fully supports the evolution feature and the due to implementation based on the log based and multi-version b-trees they extend a partial support to granularity.

3.1.6 Biological Sciences Collaboratory (BSC)

Chin and Lansing [CL04] developed the Biological Sciences Collaboratory which supports the sharing of scientific data while taking into consideration about the context. BSC captures the context in which the scientific sharing of data takes place. The point of interest in our research is that the BSC is able to track back the original source of data through the data provenance. The tracking back of data set is nothing but tracking down the evolution tree of how the data changed over time, which in fact is the evolution of data. They have also included the new edits to the document as part of the document rather than making another copy of the entire document, but are effective to only certain extent. The BSC extends partial support to the hierarchy of parts, evolution and granularity. It also supports inclusion of document partially.

3.1.7 CoEd: VTML for tracking changes

Bendix and Vitali [BV99] used the VTML to develop a system CoEd, which is a tool for creating shared structured documents for multi-user writing. The students usually work in teams during the project time which usually ends with a project report and students had problem in creating the documents and each student dealt with a part of the project which are independent and hence collaborating their work in the form of a document was tedious. The CoEd system was developed with the view to track the evolutionary changes that happen in the document by keeping in mind the need for collaboration. The CoEd accepts the file and checks the file, parses the latex code and constructs the hierarchical structure. It also supports the chapter wise editing of the single document and hence the file is broke down into smaller parts. It extends a partial support to the hierarchy of parts, granularity and document part inclusion. The CoEd is also able to extend full support to the evolution because of its version system nature.

3.1.8 Microsoft OneNote

The Microsoft One Note is a planner and note taking software from Microsoft. It is having this unique feature where in which whenever you copy over content to the one note document, the source gets automatically pasted in the system. So the information as to where the information comes from gets into the document without any manual intervention. The One Note document does not follow the principles of transclusion as the file size grows when including data from other sources.

3.2 Conclusions from related works

This section is the summary of the above section to provide a crisp image of what was discussed with respect to the four aspects. In the above section we discussed each study with reference to the four aspects which we find would be characteristics of the proposed system.

	Hierarchy of parts	Evolution	Granularity	Including parts of docs
Versioning System CVS,SVN, TFS	+	++	-	-
DEVAC	+	++	-	-
Krottmaier and Maurer [KM01]	-	+	-	++
Kolbitsch and Maurer [KM06]	-	+	+	++

CVFS	-	++	+	-
BSC	+	+	+	+
CoEd	+	++	+	+
OneNote	?	++	+	?

Figure 3.1: Evaluation of Related Study

++ supported, + partially supported, - not supported,? – No information

The table given above shows an overview of our evaluation of related work. We observe that the two systems CoEd and BSC are the systems which are the closest ones to our desired design when considering Hierarchy of parts, Evolution, Granularity and including parts of docs as the comparing features. No single system exists that fulfils all our requirements as described in the beginning of this chapter.

3.3 Summary

In this chapter we looked at various related works and evaluated them to the following aspects:

- Hierarchy of parts
- Evolution
- Granularity
- Including parts of documents

Furthermore we compared each work with the above aspects so that a much clear idea about the implementation of Data object and Information objects can be unveiled.

4 Information and Data flow analysis

In this chapter, we discuss about a case where collaboration was involved and about the problems they faced in the team. The main aim of this chapter is to develop the concept of evolution tree and a refining the concept of Information objects and Data objects. The scenario in Figure 1.1 was not sufficient to take a deep look at how data and information flows in a collaborative environment. So we decided to look a case where real life collaboration actually took place.

The three participants are faculty members at the Computer Science Department, University of Waikato. Their names are changed for anonymity. Mary, Peter and Paul started writing an article for publishing and they were collaborating among themselves to write up the article and make changes to it. Here we describe about the problems they faced while collaborating. This is written from each author's perspective so that we get a clear picture of what actually happened.

4.1 Situation as described by participants

4.1.1 Paul's View

- Paul feels that he is not the responsible author and so he did not create a versioning system for the each update

- Paul just kept the last copy. He deleted the previous copy of the file when the required changes was done and send back to Mary.

- Paul had problems when it came to pictures. Mary used a different image editor and hence when Paul was required to update a picture, he had to start building up the picture from the scratch and do the required changes.

- There were no problems with formatting for Paul, as he always used the last version sent to him by Mary.

- Paul always incremented the number of the document name which denotes the version of the document, whenever there was an update from his part.

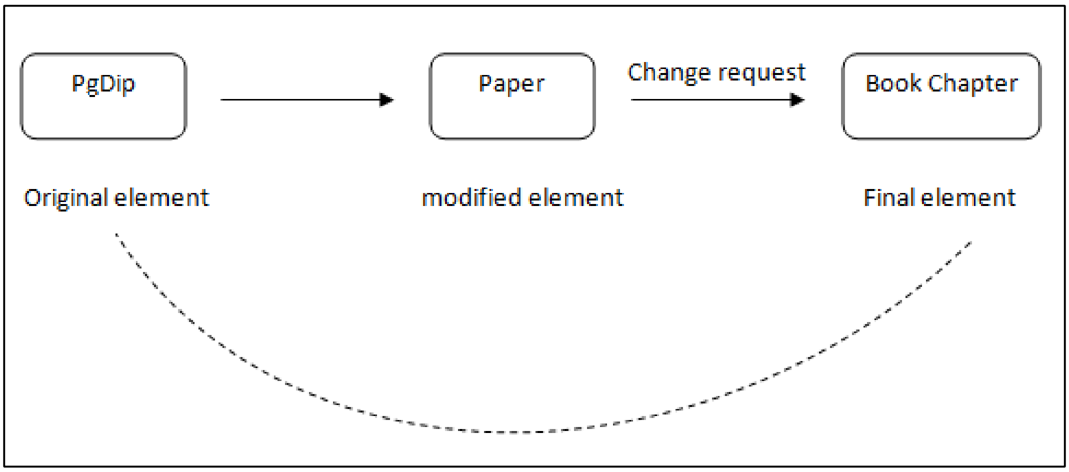


Figure 4.1: Evolution during collaboration (Paul)

From the above representation, Paul had the original document which he used in his PgDip programme and he always retained it since he was the owner of the document. The final element, which is the point of interest to us, is the modified version of his original work. Whenever a change was requested, Paul always used his original work and made all the changes due to the use of different image editor's used by the both of them and hence the logical connection between the original and final element.

During the collaboration, since Paul always made changes from his original work, he was unable to track the relation between each change as the changes were semantic. It is very difficult to track the changes as the relation is purely logical.

4.1.2 Peter's view

- Peter also did not use any versioning systems to keep track of the various versions of the document.

- There was very little email communication.

- Mary used to send the document to Peter and he takes a print out of the document and makes notes on the physical copy of the document.

- Mary and Peter then used to sit together for a discussion and Mary changes the document during the discussion

From the above, it is clear that Peter also have similar problems to Paul. Peter used to discuss with Mary and she used to make the changes to the document. Peter was active in the collaboration but he always made his changes to the physical document. This also is very difficult to track as the relation is conceptual.

4.1.3 Mary's View

- Mary always kept all the versions of the document

- Whenever a changed image was received from Paul, Mary copied and pasted the image into the work.

- To identify the various versions a gradually incrementing number system was used

- Various copies of the same image elements in different name was found in the working directory

Mary was the only one who kept all the versions. Both Peter and Paul assumed that she took all the responsibility of the paper. Mary was having problems in keeping all the versions and backing the versions to send to Paul whenever she needed a change in the picture files used in the paper.

Now, before moving further we would like to take a look at some possible generic changes that we can identify in the case study. These changes can happen to a data element and in this case we take an image element

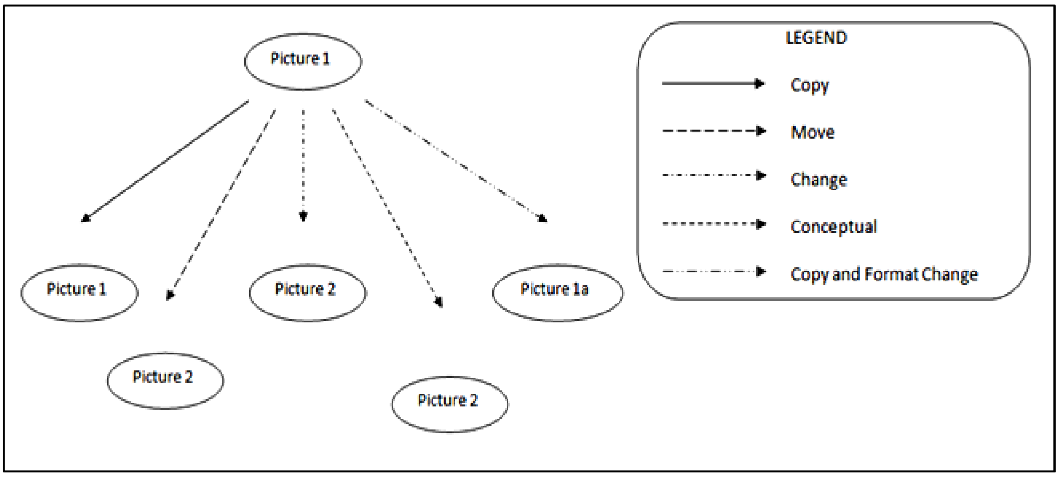


Figure 4.2: Various Possible Evolutionary methods during collaboration

The **copy process** is one in which the file is just copied over and hence there is no change to the file. The version of the file also remains as such as there is no change that is happening to the file. In the case study the copy process can be represented as the one where Paul sends over the file to Mary and Mary Just copies over the file. There is no change that is happening to the file in there.

The **Move Process** is one in which moved over. In this case there is a change that is happening to the file and hence we change the version of the file. Looking at the case study when Peter receives the file he actually takes the print out of the file and then make the changes to the file in the physical form and then both Mary and Peter sit together for a discussion in where Peter put forth his suggestion and Mary changes the file while in the discussion.

The **Change Process** is one in which there is a change that is happening to the file. In this scenario we actually change the version of the file to denote the change. In the case study Paul actually changes the picture file available in his PgDip paper upon request from Mary to suit the needs of the paper they are writing. Since there is a change that is happening to the original file we denote the change by a version change.

The conceptual relation is one in which there is no established relation between two files but there is a logical or conceptual relation between the files. Since there is a change between the files, even though it is a conceptual one, it is represented by a change in the version. Looking at the scenario, Paul kept his original picture file of his PgDip paper which was sent to Mary and when Mary requests a change to the picture, even though it is easier to make changes to the existing picture file, he always made the change to his original file due to the difference in format of the picture. So the changed file he provided to Mary is having a conceptual relation as the file is supposed to be derived or changed from the latest version of the file but Paul made the change to earlier version of the file to provide the latest version of file as he felt it as an easy way to work at it.

The Copy and change Format is one where the files are copied over and a change of format to the file takes place. The Copy and Change format is represented as a minor version change. In the case study Mary receives the picture file from Paul which is then copied over and a format change happens to the file so as to import the picture to the paper they were writing. Since there is a format change that was happening we denote and represent it as a minor version change.

4.2 Case Study Explained

The Pic 1 (refer Figure 4.3) is the picture object that was originally available with Paul when he did his PgDip. This object was needed to be used in the paper they were writing and so Paul sent the picture to Mary. Mary then copies and changes the format of the picture so that she can use the picture object in the research paper. The problem was the use of two different picture editors and the pictures that Paul sent was not compatible with the one Mary was using. So she had to change the format of the picture to use it in the paper.

After that Mary makes minor changes to the picture object but the text part remains the same. She then sends one picture to Paul with all the changes she need and sends the document File1 to Peter.

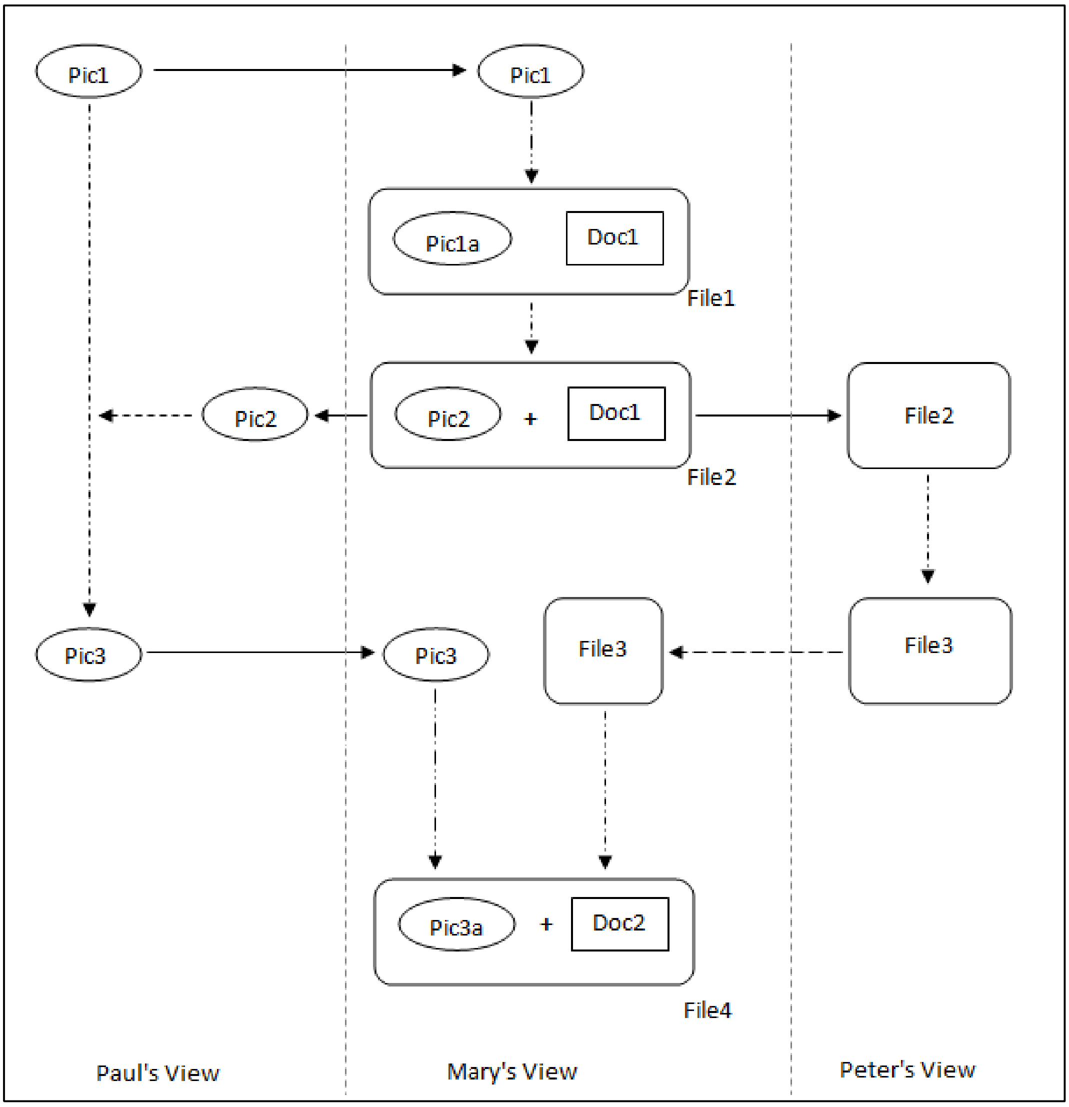


Figure 4.3: Evolution during collaboration – File Based (Whole Process)

Paul looks at the changes and then starts to make the change from the original file which he used in the PgDip. The reason for the difference in approach is the difference in file format used by each one of them. Peter on the other hand takes the print out of the file and then marks the changes in the physical file and then discusses with Mary before making any changes to paper and Mary subsequently changes the paper after the discussion.

Mary recieves Pic3 from Paul and she copies the file. The changed document constitues the pic3 and the text content, which was changed by Mary after discussing with Peter. This is the collaboration process that they followed while writing this paper.

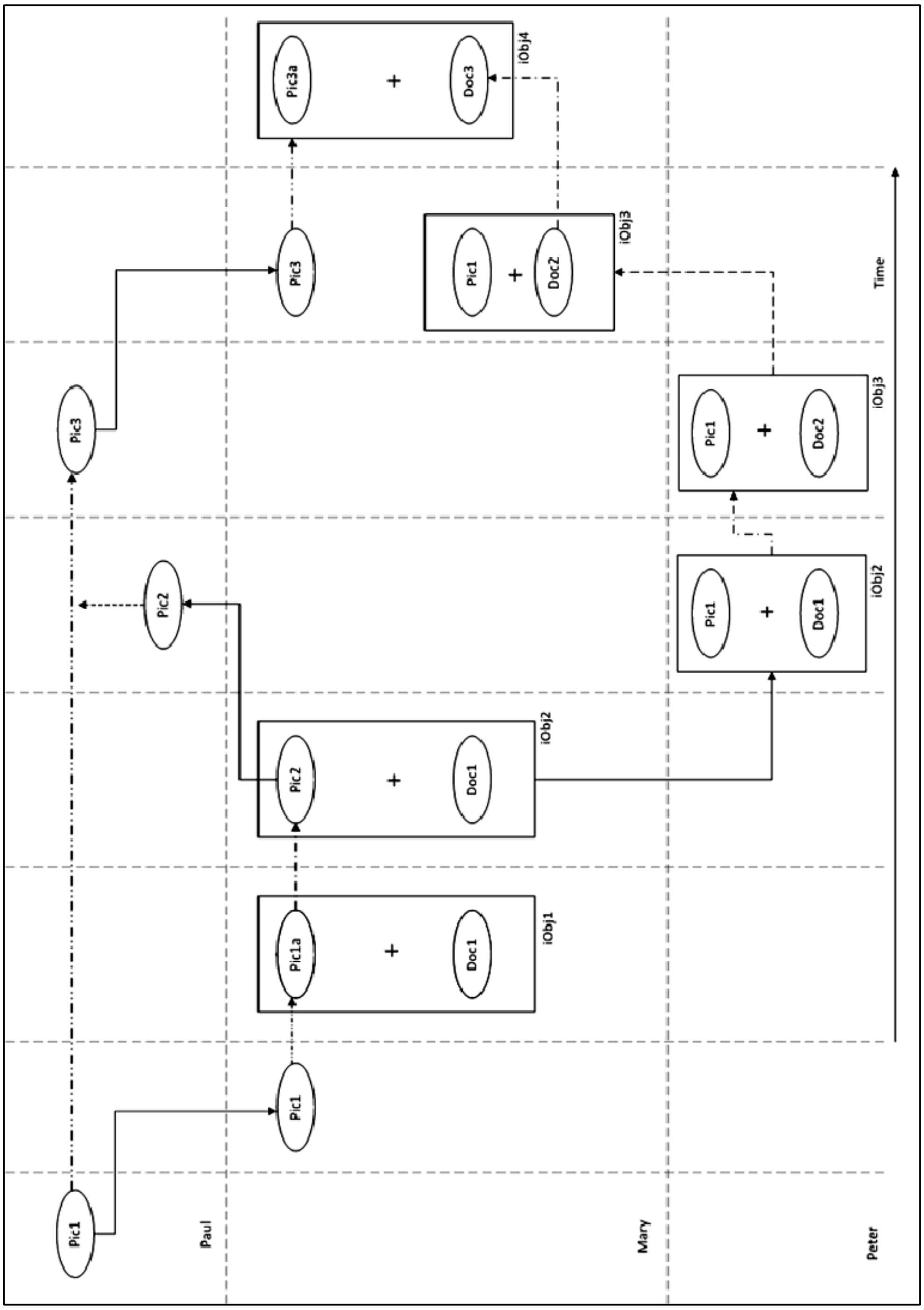


Figure 4.4: Evolution during collaboration – Conceptual

4.3 Analysis of case study

In the earlier section we have seen the case study in two ways:

- File system based (refer Figure 4.3)
- Conceptual (refer Figure 4.4)

The file system based approach (refer Figure 4.3) represents the current document concepts expect for the fact that current document concepts does not take into consideration the parts involved in but considers the file as single distinct entity. As a result we are unable to track the evolution that happens to the document as we move along the timeline.

The conceptual approach (refer Figure 4.4) represents the concept of Information object and Data object. You can also see that the conceptual approach describes and conveys the scenario much better than the File based approach. Moreover, it will be able to track the evolution by carrying the evolution information by signifying the parts of document rather than considering the document as a single distinct entity.

4.4 Summary

In this chapter, we have explained how data and information flow happens in a collaborative environment. We further looked into the concepts of Information objects and Data objects by analysing the case study that happened in a collaborative environment. We were also able to analyse between the current document concepts and the proposed concepts of Information objects and Data objects through the case study. It is clear that the evolution information is lost when the users employ the copy and paste operation on the document and we will develop the software so that the evolution information is contained even in copy and paste operations.

5 Design

In this chapter we propose the design of a system that will be able to track the evolution in the document by capturing the copy and paste of the user and thus effectively able to track the evolution of the documents, as discussed earlier in the requirements (refer section 2.4). In chapter 2 (refer Figure 2.1) we introduced the concepts of Information object and Data object. We developed the concept further in Chapter 4 (refer figure 4.4), where we introduced the conceptual view of Information object and Data object. The main aim of this chapter is to introduce the architectural diagram and the sequential process diagram of the proposed system.

5.1 Components of Proposed System

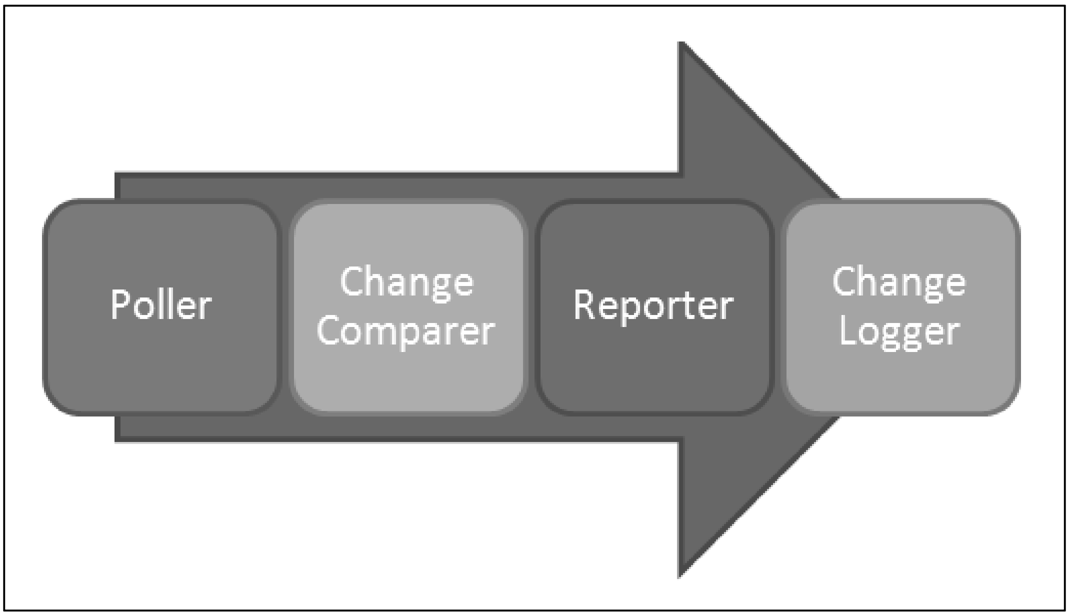


Figure 5.1: Sequential Process of proposed system

As we can see, the proposed system is mainly comprised of 4 parts

1. Poller
2. Change Comparer

3. Reporter
4. Change Logger

Now let us try to define each part of the proposed system in detail. In the proposed system, we assume that the users specify a folder as the working folder so that the system knows where you are working with your multiple documents.

Poller is a service to monitor a specified working folder and scans for any change that is made to the working folder. It is a service that is running in background. So whenever there is a change in the working folder the Poller will be able to detect the change and pass the information to the next level.

Change Comparer is an event-driven service. In the event that the Poller detects any change in the working folder, the Change Comparer tries to detect what the changes are and whether the changes need to be passed on to the next level. The main challenge of this module is the algorithm which is smart enough to do the task.

Reporter is also an event-driven module in the system. The main task of the Reporter module is report some prescribed events to the next level or the Change Logger. When the Change Comparer passes some events to the Reporter it simply passes the event with the reference to the Change Logger.

Change Logger is a logging module which logs all the events reported to it. The logs include, but not restricted to, all changes that happens to the working folder and the changes that are made to the document while the document is in the working folder.

5.2 Detailed Architecture

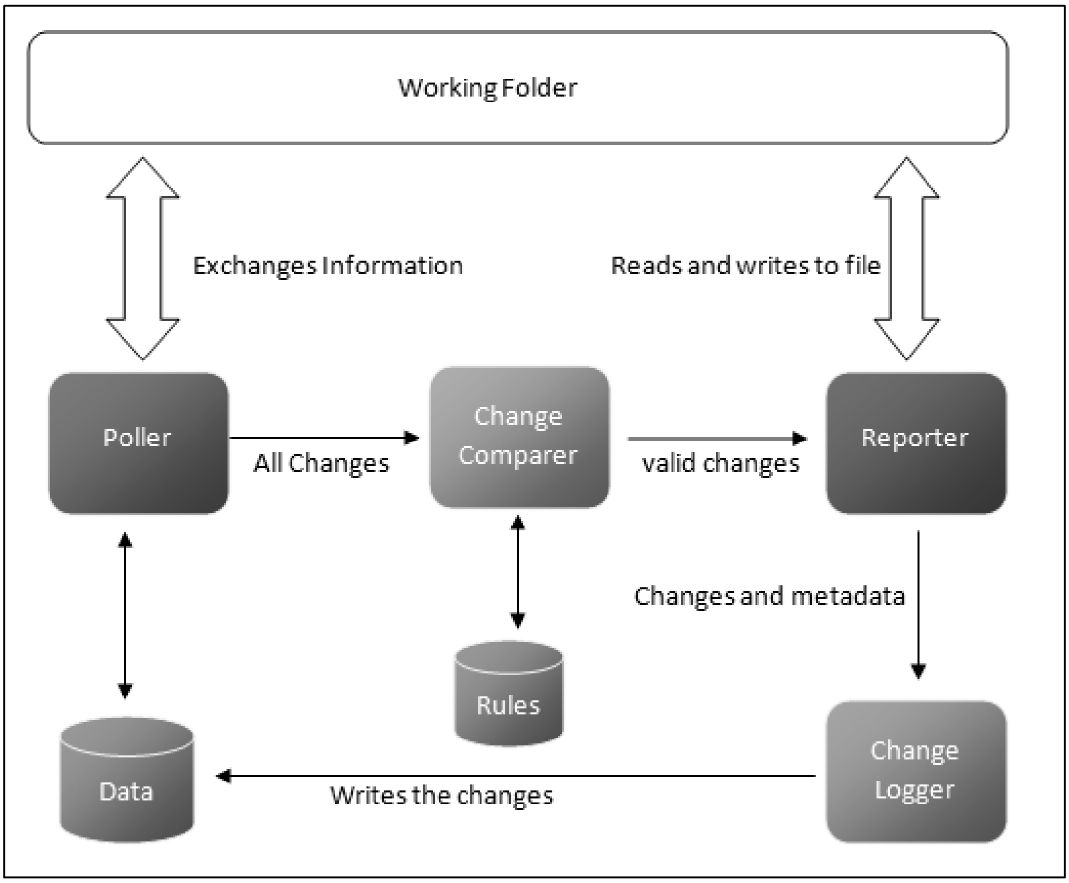


Figure 5.2: Detailed architecture diagram

We allocate a folder known as the working folder where all the documents and the resources will be placed. The working folder is necessary so that we can assign the Poller service to poll that particular folder rather than scanning the entire system for changes.

The Poller pulls the data out of the main database where all the changes are kept. The Poller then passes on all changes that it encounters to the change comparer. So whenever it detects a change in the working folder it passes on the information to the change comparer. From the figure 4.3 when Mary changes the pic1a to 2 the Poller would detect the change and pass the information to the Change comparer.

The Change comparer has a Rules Engine with which it interacts and whenever there is a valid change it passes on the information to the reporter module. So when the Poller passes the information that the pic1a is changed to pic2, Change comparer interacts with the Rules Engine and confirms that it is a valid change and subsequently passes the information onto Reporter.

The Reporter has the function of reporting the changes along with changes in metadata, if required, to the Change Logger. In our case it will report that a change has been reported from the pic1a to pic2 to the Change Logger.

The Change Logger has the function of logging all the changes that is reported to it. The Change Logger writes all the data to the main database. From the case study Change Logger will write that a change has been made to the picture file from 1a to 2.

5.3 Summary

In this chapter, we explained the sequential diagram of the proposed system. We also described the four components of the system:

- ✓ Poller
- ✓ Change Comparer
- ✓ Reporter
- ✓ Change Logger

We also looked at the detailed architecture diagram of the system and described how the system would work.

6 Implementation

In this chapter we describe about the implementation of the system. In Chapter 5 we saw the design of the system along with the detailed architecture diagram of the system. The main aim of this chapter is to describe how the components were implemented. We have developed the system on C# programming language with SQL Server 2005 as the backend. The following are the four components of the system:

- Poller
- Change Comparer
- Reporter
- Change Logger

We will then try to answer the following questions:

- How does the system work
- What were the challenged faced while developing the system
- How were those challenges faced and tackled

6.1 Poller Component

While considering the implementation of Poller Module we looked into various clipboard software utilities to understand the working and to extend the capabilities to meet our requirements as outlined in the requirements (refer section 2.4). We also wanted the software to be open source so that it will be easy to modify the code and to extend it so that it meets the requirements of the proposed system.

The first system that we looked was the clipboard management utilities, because the Poller module is required to catch the changes made to the working folder on the fly. We looked at clipboard management utilities because the most common way of changing the file is either copy and paste or cut and paste or just highlighting the text allowing it for drag and drop. The Clipboard is the place where data resides during the first phase of the operation.

6.1.1 Situation & Requirements

The following points indicate the scenario in which we evaluated the third party software we came across. The situation that we use is from the case study that we had discussed earlier. From the case study, the following are the sequential representation of what is happening

1. Paul sends the picture (pic1) file to Mary
2. Mary receives the picture (pic1) file
3. Copies the picture (pic1)
4. Paste picture (pic1) to her picture editor
5. Save the picture (pic1a) in a different format.

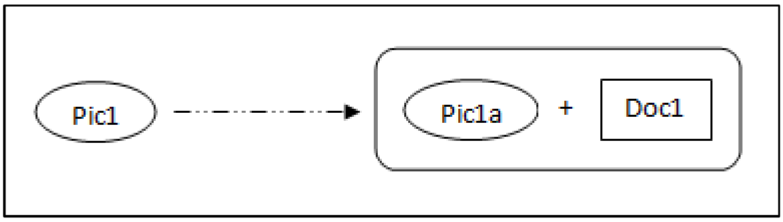


Figure 6.1: Situation example

- There are two files pic1 and pic1a which are open in the file system
- Copy of content from pic1
- Paste content of pic1 to pic1a

The requirements are the set of functionality that we would want from the proposed system and which would in a way help in evaluating the third party software.

- The physical path, the place where the file resides in file system, of pic1

- The Physical path, the place where the file resides in file system, of pic1a
- The copied content from pic1

6.1.2 Evaluation Criteria

We will evaluate the third party software based on the following criteria. These criteria are an extension of functionality from the requirements of evaluating the software. We would like to put forth these criteria as benchmark while going through the process of evaluating the third party software

- **Detect Copy Process -** This refers to the ability of the software to detect the copy process whenever it happens in the system. For example, when the user copies the content from pic1, the software should be able to detect that copy action from the user.
- **Detect Copied Content -** This is the ability of the software to get the content that was copied. For example, when the user copies the content from pic1, the software should be able to detect what was copied from the pic1.
- **Source File Window -** It refers whether the software can track the Window name of the source from which the content was copied. For example, when the user copies content from pic1, the software should be able to get the source window as "pic1 – pictureeditorprogram" if the pic1 is edited by pictureeditorprogram.
- **Source File Path -** It refers whether the software can track the file path of the source where the copy process happened. For example, when the user copies content from pic1, the software should be able to get the physical location of the pic1.
- **Destination File Window -** This is the ability of the software to track the Window name of the destination to which the content was transferred. For example, when the user copies content from pic1 and pastes it to pic1a, the software should be able to get the destination window as "pic1a – pictureeditorprogram" if the pic1a is edited by pictureeditorprogram.

- **Destination File Path -** This is the ability of the software to track the file path of the destination when the paste process happened. For example, when the user copies content from pic1 and subsequently paste the content to pic1a, the software should be able to get the physical location of the pic1a.

6.1.3 ClipMagic Lite

"ClipMagic Lite", commercial software from MJT net [CML] which was able to give you the window name from which the data was copied from. This information is very crucial for our project. Moreover this software was able to give the physical file path of certain kind of file such as Microsoft word. This means that, if a textblock was copied "ClipMagic Lite" is able to give the source as to where the file resides in the file system. This however is commercial software and hence the source code was not available for modification. While evaluating this software the following was the criteria that were satisfied. The software was able to

- ✓ Detect copy process
- ✓ Detect copied content
- ✓ Detect source file window
- ✓ Detect source file path (for some file types)
- × Detect destination file window
- × Detect destination file path
- × Open source

6.1.4 Easy Clipboard Manager

"easyclipboardmanager"[ECM] is an open source project which does the clipboard management operations . The easy clipboard manager is able to capture the user events such as copy & paste, drag & drop and cut & paste. They have achieved this by loading the windows kernel and user Dynamic Link Libraries (DLL) and configuring the entry point to the DLL as copy operation. So the code will be injected to the DLL when the entry point matches the user operation which is, data being placed to the clipboard. We evaluated this software based on the criteria and found that the software was able to

- ✓ Detect copy process
- ✓ Detect copied content
- ✓ Open source
- × Detect source file window
- × Detect source file path
- × Detect destination file window
- × Detect destination file path

6.1.5 Easy Clipboard Manager++

The Easy Clipboard manager++ is the extension of the Easy Clipboard Manager software. Since it is an open source project we were able to modify the code and make it work to meet our requirements.

As the part of the requirement we were required to detect the source file window name. This information is very vital to track down the source from where the content was copied. To achieve this we had to inject the DLL to find out the active window of the user when the copy action takes place and then make the function call to get the window text which is an in-built function that the DLL provides through the .NET Framework.

The source window title could also be tracked down by injecting the DLL to get the process identifier when the copy action takes place. With the process identifier, it is possible with the help of System.Diagnostics which is a namespace in .NET framework which allows working with system process, to get the Filename of the process.

The poller component is a very crucial part of this software and we will spend more time in implementing this component. The implementation of the poller component faced many challenges. We will detail the implementation by stepping into each of the challenge that we faced while developing this component.

6.1.6 Challenges

In this section we list out the various challenges that we faced during the development of the system.

- Tracking the copy process
- Tracking the source file path
- Tracking source window
- Tracking paste process
- Tracking destination window
- Tracking destination file path

These are the various challenges that we faced while we developed the system. We will now discuss how these challenges were tackled so as to implement the system.

6.1.7 Tracking the copy process

In order to track the evolution of documents, it is imperative that we track where the contents come from. Copy process is one such method where the users copy the content from one place to another. In a collaboration, refer Figure 4.2, users tend to use the copy process in order to transfer the content from one place to another.

6.1.7.1 Problem description

We were required to monitor the copy process by the user. When the user tries to copy the content from one place to another our application need to be notified that there is a copy process happening

6.1.7.2 Solution

We figured that clipboard is a place where all the copied items were stored. So we began looking for an open source project which can keep track of changes to the clipboard. We then found out a project "easyclipboardmanager" and then made changes to the project to meet our requirements.

When the user does a copy process, the content for the clipboard changes and the .NET library provides an event whenever there are changes to the clipboard. By continuously monitoring the clipboard we were able to track the copy process by the user in a system. We were able to do this by importing the user32.dll with the clipboard change event which triggers whenever there is a change to the clipboard.

6.1.8 Tracking the source file path

In the previous section we explained how the copy process of the user could be tracked. The next challenge was to get the source file path. Source file path is the file path or the URL of the content. When the user copies content from a place A to place B, source file path is the token that identifies the location of A. It can be a file path in the physical system or a URL of a website.

6.1.8.1 Problem description

We require the location of the source from which the content actually came from. For example, when a user copies content from document A to document B, we are required to track the location of document A.

6.1.8.2 Solution

To address this problem, we first took a look a path where we grab the process id of the process which made the copy process and then somehow track the file path from the process. But to continue down that path we were required to load all the DLL's so that the file type could be recognised by the .NET framework. We found this method to be programmatically expensive and dropped this.

We then took a look at how data was actually stored in the clipboard. We figured that the data in clipboard when it is a common file type was stored in HTML format with headers. We found a SourceURL flag in the header which point to the file path or URL of the location.

Suppose we copy some content from 'www.somewebsite.com' to one of the document in the system, the clipboard stores the data like

```
Version:1.0
StartHTML:000000183
EndHTML:000008771
StartFragment:000008482
EndFragment:000008631
StartSelection:000008482
EndSelection:000008631
SourceURL:http://www.somewebsite.com/
...html content...
```

We were able to programmatically extract the SourceURL flag from the header in the clipboard and thus address the issue.

6.1.9 Track source window

The source window name is the name of the window from which the content was copied from. In the previous section we tracked the source file path but it is limited to some known file types. In other words the header in the clipboard will carry the SourceURL flag if the content copied is of HTML data format. To overcome this limitation we figured that source window name should be tracked so that we can get a good idea where the content came from.

6.1.9.1 Problem Description

The application was required to track the source window name when there is a copy process event is triggered. For example, if the user is copying from a Microsoft Word document "somename.docx" then the window name of the document when copy process takes place would be "somename.docx – Microsoft Word".

6.1.9.2 Solution

The solution was based on the fact that, when the user does a copy process from a document or window the window should be the active one and currently the one with the focus.

With this fact we looked into the methods that the "user32.dll" of the .NET library provides and we figured that there are methods to get the foreground window at any given time which would return a handle to the active window. The foreground window is the window which is currently in focus. We then feed this handle to another method to get the window text which effectively is the source window name.

We were able to get the source window text once we moved these methods to the event where we track the copy process and then we were able to get the source window name.

6.1.10 Tracking the paste process

The tracking of evolution of document cannot be complete without tracking the paste process. Paste process is the event where the user transfers the data to the destination. In the previous section we saw how the copy process and tracking various source locations was addressed. In this section we will see how the paste process of the user can be tracked.

It is difficult than tracking the copy process because when the user copies some content there is an event trigger. We were able to track the copy process by monitoring the event which triggers when there is a clipboard change. The fact that made the tracking of the paste process difficult is that there is no event that is triggered when the clipboard was accessed.

6.1.10.1 Problem description

We were required to track the paste process when the user performs a paste action. The fact that there was no event that was triggered when the user accesses the clipboard would require us to figure out a workaround to track the paste process.

6.1.10.2 Solution

The problem was addressed by the use of global hot keys. Global hotkey is a key combination, which is application specific, which can be used to perform a user action quickly. The paste process has an already assigned hotkey combination in windows OS. The key combination is CONTROL KEY + V.

The .NET framework allows us to override the key combination which is already assigned for the paste process. Since we did not want to re-calibrate the standard key for the paste process, we decided to override and make it available to our application.

When the application loads we register the CONTROL KEY + V as our global hotkey. The user then performs the paste process by using this key combination. During the process the hotkey sends a system message to our application. We then un-register the hotkey so that we can send CONTROL KEY + V to the current application in focus and complete the paste process.

The only limitation of this method is that it will not be able to incept the paste process when the mouse is used and will require the user to initiate the paste process by keyboard.

6.1.11 Tracking destination window

The destination window the name of the window to which the content which is copied by the user is being transferred or pasted. In other words, it is the name of the window which initiates the paste process.

6.1.11.1 Problem description

We are required to get the window name of the window which initiated the paste process. For example, the user does a copy of content from one word document “A” and then initiates the paste process to another word document “B”. We needed to get the window name of the destination, that is, window name of the word document B.

6.1.11.2 Solution

As discussed in the previous section, the paste process is not having any events and as such no headers to look into. The global hotkey is the method by which we tracked the paste process.

We decided that we could use the technique of getting the handle of the window which is currently in focus. This handle when supplied to get window text method of the “user32.dll” of the .NET library gives us the window name of the window which initiated the paste process.

When the user initiates the paste process, it sends a message to our application. By incorporating the above method of getting the window text to the method when the application receives the message that paste process is happening solved the issue of getting the window text of the destination window.

6.1.12 Tracking the destination file path

In this section we will explain how we managed to get the destination file path when there is a paste action. Destination file path is the physical location of the destination to which the user transfers the copied content from the clipboard. Our solution is based on the fact that we require the user to assign the working directory so that we can track changes to that particular directory.

6.1.12.1 Problem description

We were required to get the physical file path of destination. When the user performs a paste action, the user transfers the data to a destination and we need to find the physical location of that file.

6.1.12.2 Solution

The fact that we require every user to set a working directory negates the need to monitor the whole system. In the previous section we managed to get the destination window name of the window which initiates the paste process. The window name of a sample word document would be “somename.docx – Microsoft Word”.

When you look at the window name we can easily get the filename of the destination document. The only thing to be done was to look for the files in the working directory that matches the window name and then append the file path of the working directory.

Once we have the file path we updated the database by using the unique identifier that we assigned to the clipboard content. It is based on the fact that a user can only paste the latest content in clipboard. So we pulled the unique identifier to the recently added content from the database and performed the update action, making us available the destination file path.

We then evaluated the modified software and found that it was able to

- ✓ Detect copy process
- ✓ Detect copied content

- ✓ Detect source file window
- ✓ Open source
- ✓ Detect source file path (for some file types)
- ✓ Detect destination file window
- ✓ Detect destination file path

6.1.13 Comparative study of Poller component

Based on our analysis on evaluating these projects we came to conclusions which is represented in the table below. The conclusions are based on the evaluation criteria we had put forth before.

	Clipmagic Lite	Easy clipboard Manager	Easy clipboard Manager^{++}
Copy Process	++	++	++
Copied Content	++	++	++
Source Window	++	-	++
Source File path	+	-	+
Destination Window	-	-	++
Destination File path	-	-	+
Open Source	-	++	++

Figure 6.2: Software analysis based on evaluation criteria

++ supported, + partially supported, - not supported

From the table 6.1 we can easily conclude that the Easy clipboard manager^{++}, the software that we modified from the original easy clipboard manager project is able to adhere to the various evaluation criteria that we had put forth towards the implementation of the system.

6.2 Change Comparer

The change comparer is the module which determines the changes detected are to be passed to the next reporter module. The change comparer work on rules and when a change is detected it checks against it rules and determines whether the changes are to reported and then pass it to the reporter. This is a simple software module where the changes intercepted are compared against a set of rules. If the change detected passes the rule which we have written it passes to the reporter module otherwise it drops the change. The simple state based UML diagram of the module is given below:

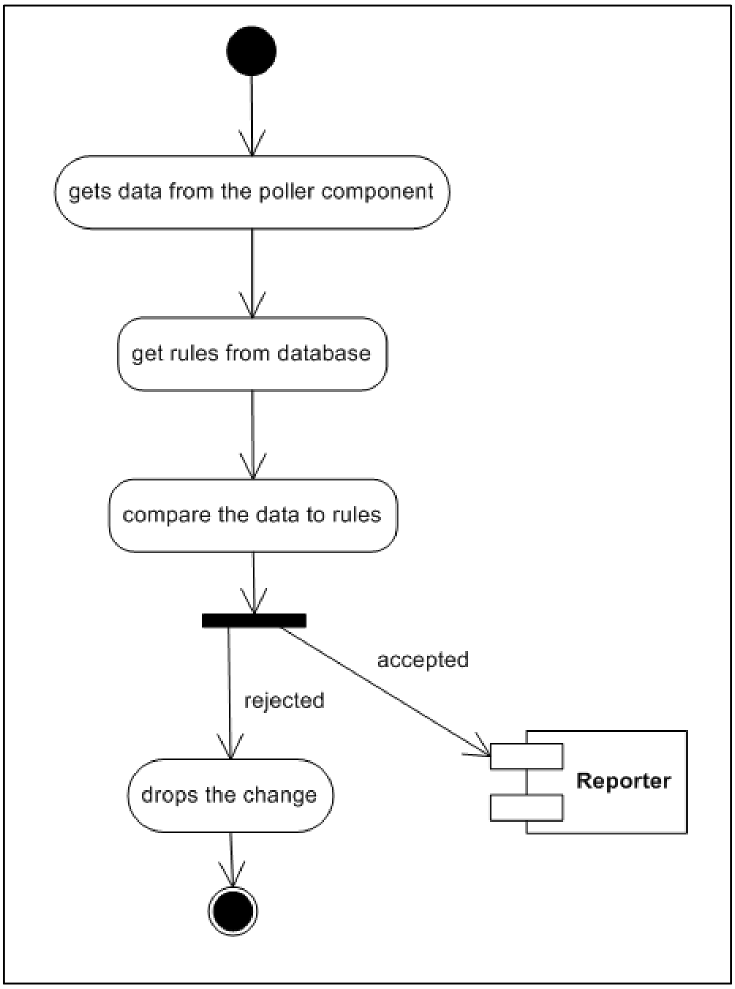


Figure 6.3: UML state diagram for change comparer component

From the figure 6.2 we can easily understand the working of this component. A step by step algorithmic explanation of this component is given below:

1. Change comparer gets the data wide a copy user action from the Poller
2. Fetches the rule from the database
3. Change comparer then compares the data to the rules
4. Passes the data to the Reporter component if it passes the rule validation or drops the change when the data fails to meet the rule validation

6.3 Reporter

The reporter module is a program which passes the change to the change logger. We have kept this module in accordance with the patterns and practises which deals which software development. Currently it passes the information over to change logger and acts as a proxy. We have reserved the reporter module mainly for future use where it can be used to tag with metadata if needed.

For the purpose to conform to the pattern and practises, we developed a class ‘reporter’. The reporter class has the following functionalities:

- ✓ **Receive message**: This functionality is used to receive the message from the change comparer.
- ✓ **Processor**: The processor is reserved for future work if there has to some change to the data that is passes like changing the metadata.
- ✓ **Send message**: The send message functionality is used to send the data that the reporter component receives to the Change logger component.

6.4 Change Logger

The change logger deals with logging the changes that is passed to it to the database. This also is a simple module which deals with connecting to the SQL server 2005 and then logging the changes. It mainly performs INSERT and UPDATE operations to the concerned database.

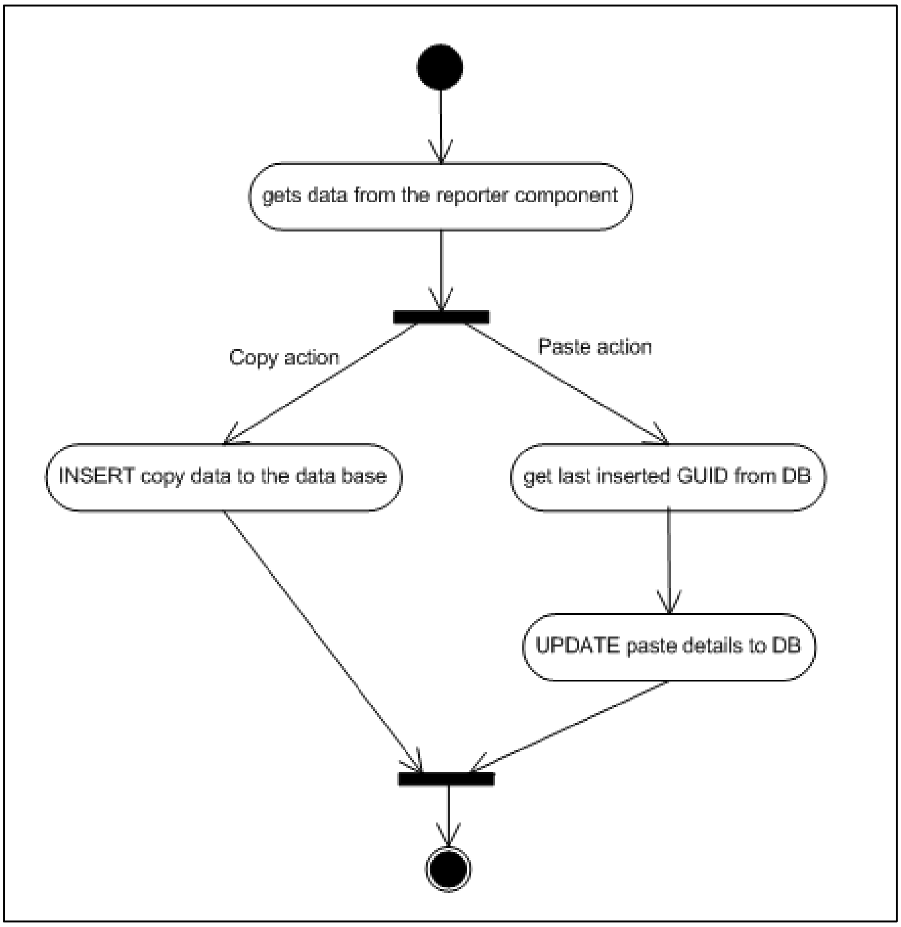

gets data from the reporter component
Copy action
Paste action
INSERT copy data to the data base
get last inserted GUID from DB
UPDATE paste details to DB

Figure 6.4: UML state diagram for change logger component

The UML state diagram of the change logger is given by figure 6.3. To understand the working of the component more clearly, we will now explain the step by step process of what is implemented.

- The change logger gets the data from the reporter
- Based on the user action the change logger takes different action routes. The two user actions are:
  - ➢ Copy action
  - ➢ Paste action
- In case of copy action, the change logger INSERT the data directly into the database
- In case of paste action,

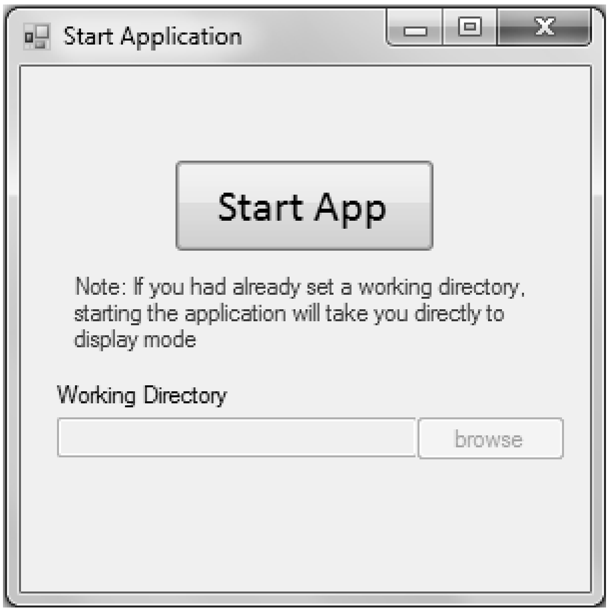
- ✓ The change logger gets the unique identifier for the most recently inserted content because of the fact that the paste action provides you with the latest addition to the clipboard which in fact is the latest entry to the database since we monitor the clipboard and insert the data to our database.
- ✓ It then UPDATES the data base the associated paste details with the help of the unique identifier thereby associating the copy and paste actions.

6.5 Working of the system

In this section we describe the working of the system. We will describe the working of the system by providing the screenshots and then explaining each screen shot in detail.

6.5.1 The application start window

The Start window prompts the user to start the application. One they start the application it will prompt for a working directory, that is, the directory which the user wants to be monitored.



Note: If you had already set a working directory, starting the application will take you directly to display mode

Figure 6.5: Starting Application

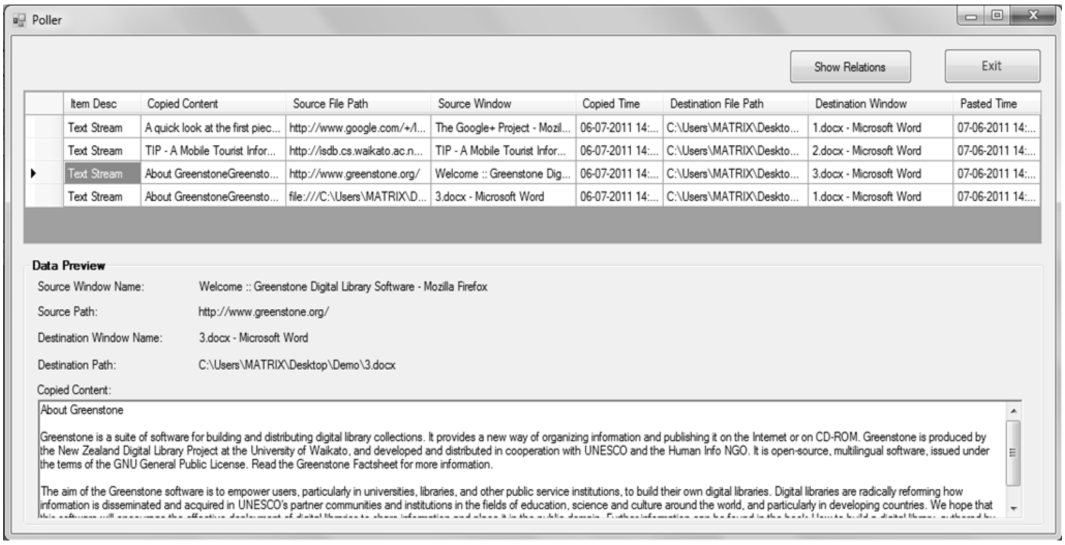
If the user has already specifies a working directory for the system, it will directly take the user to the activity window where the working directory will be monitored in real-time.

When the user runs the application for the first time, the application logs the system name along with the selected working directory of the user. Whenever the user tries to start the application for the second time it fetches the information from the database and compares it to the system name to by-pass the working directory prompt to the user.

6.5.2 Poller activity window

The poller activity window presents the real-time user updates. The following are the user actions that are captured by the poller:

- ✓ Copy of data
- ✓ Paste of data



	Item Desc	Copied Content	Source File Path	Source Window	Copied Time	Destination File Path	Destination Window	Pasted Time
	Text Stream	A quick look at the first piec...	http://www.google.com/+/l...	The Google+ Project - Mozil...	06-07-2011 14:...	C:\Users\MATRIX\Deskto...	1.docx - Microsoft Word	07-06-2011 14:...
	Text Stream	TIP - A Mobile Tourist Infor...	http://isdb.cs.waikato.ac.n...	TIP - A Mobile Tourist Infor...	06-07-2011 14:...	C:\Users\MATRIX\Deskto...	2.docx - Microsoft Word	07-06-2011 14:...
▸	Text Stream	About GreenstoneGreensto...	http://www.greenstone.org/	Welcome :: Greenstone Dig...	06-07-2011 14:...	C:\Users\MATRIX\Deskto...	3.docx - Microsoft Word	07-06-2011 14:...
	Text Stream	About GreenstoneGreensto...	file:///C:\Users\MATRIX\D...	3.docx - Microsoft Word	06-07-2011 14:...	C:\Users\MATRIX\Deskto...	1.docx - Microsoft Word	07-06-2011 14:...

Data Preview

Source Window Name: Welcome :: Greenstone Digital Library Software - Mozilla Firefox

Source Path: http://www.greenstone.org/

Destination Window Name: 3.docx - Microsoft Word

Destination Path: C:\Users\MATRIX\Desktop\Demo\3.docx

Copied Content:

About Greenstone

Greenstone is a suite of software for building and distributing digital library collections. It provides a new way of organizing information and publishing it on the Internet or on CD-ROM. Greenstone is produced by the New Zealand Digital Library Project at the University of Waikato, and developed and distributed in cooperation with UNESCO and the Human Info NGO. It is open-source, multilingual software, issued under the terms of the GNU General Public License. Read the Greenstone Factsheet for more information.

The aim of the Greenstone software is to empower users, particularly in universities, libraries, and other public service institutions, to build their own digital libraries. Digital libraries are radically reforming how information is disseminated and acquired in UNESCO's partner communities and institutions in the fields of education, science and culture around the world, and particularly in developing countries. We hope that

Figure 6.6: Poller activity feed

The poller, as discussed in the previous chapter, captures the following details:

- ✓ Copied content
- ✓ Source file path
- ✓ Source window name
- ✓ Destination file path
- ✓ Destination window name
- ✓ Copy time
- ✓ Paste time

The activity feed of the poller is presented in a data gridview format and on clicking or highlighting a particular row or column; it presents the user with the concise data preview of the particular user action.

The copy process of the user is captured by monitoring changes to the clipboard. The clipboard is the area of the operating system which keeps the data temporarily. We continuously monitor the clipboard for changes and then track the copied content by assigning a unique identifier and then storing it in our database.

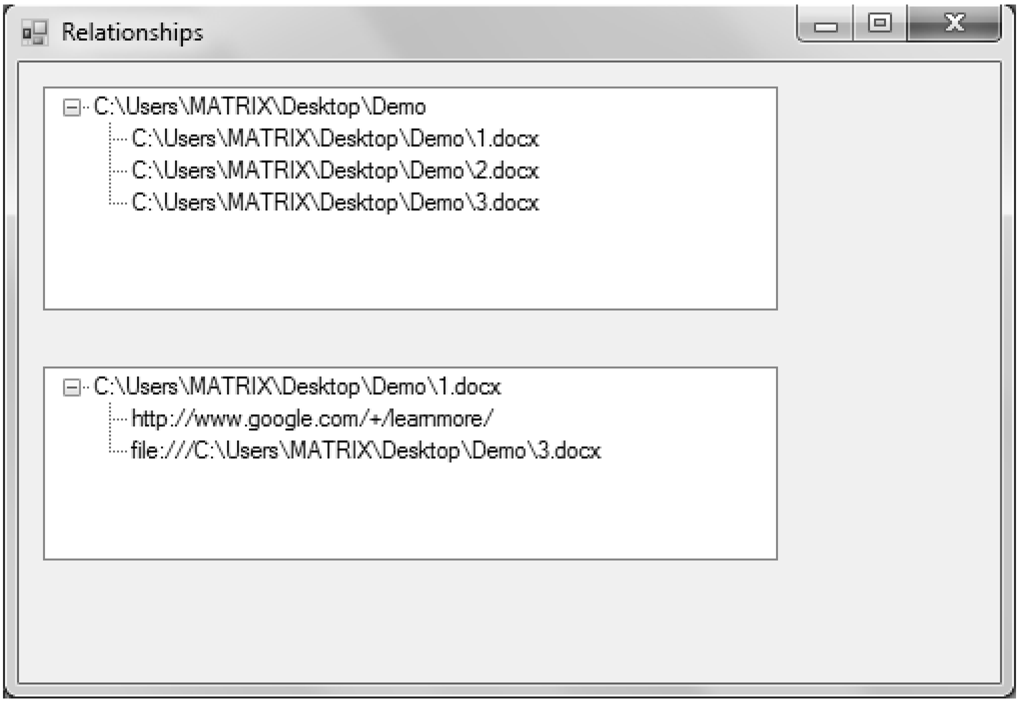
We also track the paste process of the user by assigning a global hotkey for our application. By assigning the global hotkey, we are efficiently able to track the paste process. When there is a paste process we track the copied content and then update the database with the help of the unique identifier which we assigned during the copy process.

6.5.3 The relationship window

The relationship window shows relationship of the files in the working folder. It shows the following

- ✓ Files in the working folder
- ✓ Various dependency associated with the selected file

The relationship window presents the user with a tree view kind of representation. It shows the root node as the working folder and then shows the files in the working folder as the children. To know the various relationships associated with each document the user needs to click a particular node.



```
⊟ C:\Users\MATRIX\Desktop\Demo
    C:\Users\MATRIX\Desktop\Demo\1.docx
    C:\Users\MATRIX\Desktop\Demo\2.docx
    C:\Users\MATRIX\Desktop\Demo\3.docx
```

```
⊟ C:\Users\MATRIX\Desktop\Demo\1.docx
    http://www.google.com/+/learnmore/
    file:///C:\Users\MATRIX\Desktop\Demo\3.docx
```

Figure 6.7: The relationship window

Once the user clicks on the child node, it presents the user with another tree structure where the root node is occupied by the selected file and the various sources associated with it as the children.

6.6 Summary

In this chapter we saw, in detail, about the implementation of the various components of the system. We also saw a comparative study of the most important poller component which is instrumental in implementing the system. The challenges of the poller component have been presented with a problem description which details the problem and the solution which describes the approach to solving that problem. The poller component was gradually extended from an open source project named "easyclipboardmanager" and then modified to meet our requirements as outlined in section 2.4. The working of the system is detailed with screenshots of the working system.

The implemented system conforms to two of our requirements (refer section 2.4). The implemented system is able to:

- ✓ Track the evolution information (R3)
- ✓ Evolution is tracked without any manual intervention (R4)

7 Evaluation

In this chapter, we will assess the implemented system. For the purpose of evaluating the software we will conduct an expert walkthrough. The expert group will be constituted by the people in the case study (refer chapter 4). The expert group will be presented with the case study which happened during their collaboration and then introducing the implemented system. We will then gather feedback and then recording each of their feedback.

7.1 Assessment Plan

The assessment of the software consists of the presentation of the case study to focus on the problems. The case study presentation will be followed by an explanation of the implemented system. The main of the walkthrough is to have:

- ✓ A detailed discussion about the software
- ✓ Find problems in the system
- ✓ Get suggestions for improvement

The experts will constitute Mary, Peter and Paul. The walkthrough will have two phases:

- ➢ Individual presentation to each expert
- ➢ Presentation to the expert group

In the individual walkthrough, we will present the case study and software to each expert to get their individual expert opinion and get the feedback from them. In the individual presentation we will present the expert with their part in the collaboration. We will also get the suggestion to improve the software.

The expert group walkthrough is designed to have a detailed discussion about the software to find more problems about the system so that the system can be improved. Since the participants are experts we can get more feedback about the domain too. We will then detail out the outcome of the group walkthrough.

7.2 Individual expert walkthrough – Mary

In this section we will discuss about the Individual expert walkthrough conducted for the expert, Mary. The case study presented and walkthrough for the implemented system led us to discover some usability issues. The main outcome of the walkthrough is detailed.

1. Mary was unclear about the relationship window (refer figure 6.6). Mary suggested that we put some labels so that the user would know what they are dealing with. This suggestion was incorporated into the system so as to give more meaning about the window to the user.
2. Mary also suggested that we add levels to the tree view in the relationship window (refer figure 6.6) so that the user does not have to select the file from the files that is available in the working directory display. The usability issue was also incorporated into the system so that the user would feel that the system is more user-friendly.
3. Another suggestion was to change the order of the window in which they appear. Mary felt that the user activity feed window (refer figure 6.5) should be presented with relationship window (refer figure 6.6) when the user starts the application. Mary justified this by pointing out that this will help the users to answer a possible question. This suggestion was put on hold due to time constraint as the change involves significant re-work.
4. Mary also put forward another suggestion to colour the nodes in such a way that it will indicate to the user which files were actually authored by a particular user. This suggestion was not incorporated as the File information would be available only if we open the file during process and hence a very costly, in programmatic terms, change.

7.3 Individual expert walkthrough – Paul

In this section we will discuss about the Individual expert walkthrough conducted for the expert, Paul. The case study was presented and the software walkthrough was done. The outcome of the walkthrough is detailed:

Paul was happy that the system was able to track the copy and paste actions but he suggested that he would probably not use the system. The problem was that the system does not currently address the conceptual relationship. The problem that Paul had in the collaboration was semantic and the proposed system does not cover the semantic relationship. The semantic relationship in the documents is a very complex area requiring much research and we will discuss about this in the future works.

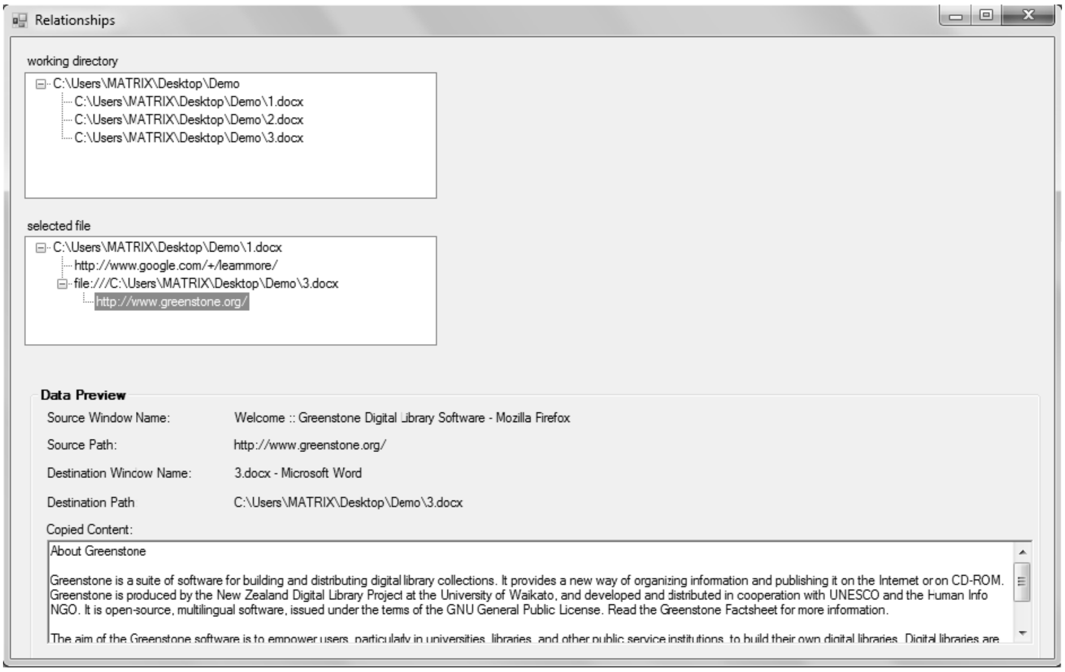
7.4 Individual expert walkthrough – Peter

In this section we will discuss about the Individual expert walkthrough conducted for the expert, Peter. The case study and the walkthrough led to the following outcomes:

1. Peter feels that he is more of a semantic person and would like to work with hardcopies
2. Peter also suggested that he would not use the system since he feels that he would be swamped data which we would not actually need and he feels that it would be difficult to find the useful data from the huge pile of accumulated data.

7.5 Improved relationship window

This section deals with explaining the re-structured relationship window following the walkthrough with Mary. We introduced the labels to provide more sense to the user and made the application more user friendly by addressing the usability issue with the relationship window. The revamped screenshot of the relationship window is given in figure 7.1.



C:\Users\MATRIX\Desktop\Demo
C:\Users\MATRIX\Desktop\Demo\1.docx
C:\Users\MATRIX\Desktop\Demo\2.docx
C:\Users\MATRIX\Desktop\Demo\3.docx

C:\Users\MATRIX\Desktop\Demo\1.docx
http://www.google.com/+/learnmore/
file:///C:\Users\MATRIX\Desktop\Demo\3.docx
http://www.greenstone.org/

Source Window Name: Welcome :: Greenstone Digital Library Software - Mozilla Firefox
Source Path: http://www.greenstone.org/
Destination Window Name: 3.docx - Microsoft Word
Destination Path C:\Users\MATRIX\Desktop\Demo\3.docx
Copied Content:

About Greenstone

Greenstone is a suite of software for building and distributing digital library collections. It provides a new way of organizing information and publishing it on the Internet or on CD-ROM. Greenstone is produced by the New Zealand Digital Library Project at the University of Waikato, and developed and distributed in cooperation with UNESCO and the Human Info NGO. It is open-source, multilingual software, issued under the terms of the GNU General Public License. Read the Greenstone Factsheet for more information.

The aim of the Greenstone software is to empower users, particularly in universities, libraries, and other public service institutions, to build their own digital libraries. Digital libraries are

Figure 7.1: improved relationship window

We introduced the labels 'working directory' to tree view which represents the files inside the working directory and 'selected file' to represented the user selected file from the working directory. The 'selected file' tree view provides the user with the source information to the file that the user has selected from the 'working directory'.

We also re-structured the 'selected file' tree view to be more user friendly by adding levels so that user can just click on the source node of 'selected file' and the source nodes will be added after querying the data base.

The data preview functionality was also added to the relationship window which allows the user to take a look at the source information by staying on the relationship window. This also adds to the usability of the system as it does not require the user to move to the user activity feed window to know about the details about the source.

7.6 The group expert walkthrough

In this section we will discuss about the walkthrough to the group of experts, Mary, Peter and Paul (their names are replaced for anonymization). The main aim of the group expert walkthrough is to initiate a discussion about the case study leading to the software so as to find the shortcoming of the software and get more suggestions to improve the system. The walkthrough was mainly discussing about the case study in detail (refer to section 4.2) and then moved on to discussing the software. Each of the experts tried to identify the problem they faced during the collaboration and then to relate it to the software to see if it resolves their problem. The walkthrough led to the following:

- Peter was happy about Mary taking the responsibility on the paper even though he contributed towards the writing and hence kept no versions of the paper.
- Paul also handed over the responsibility to Mary and he pointed that he was not sure about the role he played in the collaboration and went on keeping the last copy and master copy of the picture file involved and hence did not resort to keeping all the versions.
- Mary was not able to easily backtrack two versions since Paul kept only the last copy and she found it awful to go through all the versions whenever she needed a change from Paul.
- Mary also argued that it is easy for them by not using the system as both Peter and Paul kept no versions and assumed Mary to take the responsibility.

7.6.1 Case Study re-visited

In this section we will re-visit the part of case study which formed a part of a major discussion in the expert group walkthrough. Figure 7.2 gives us a brief idea of the major issue they faced between Paul and Mary during the collaboration to write the paper.

Paul always had only two copies of the picture, pic1 which is the master copy and either pic 2, 3, 4 or 5.

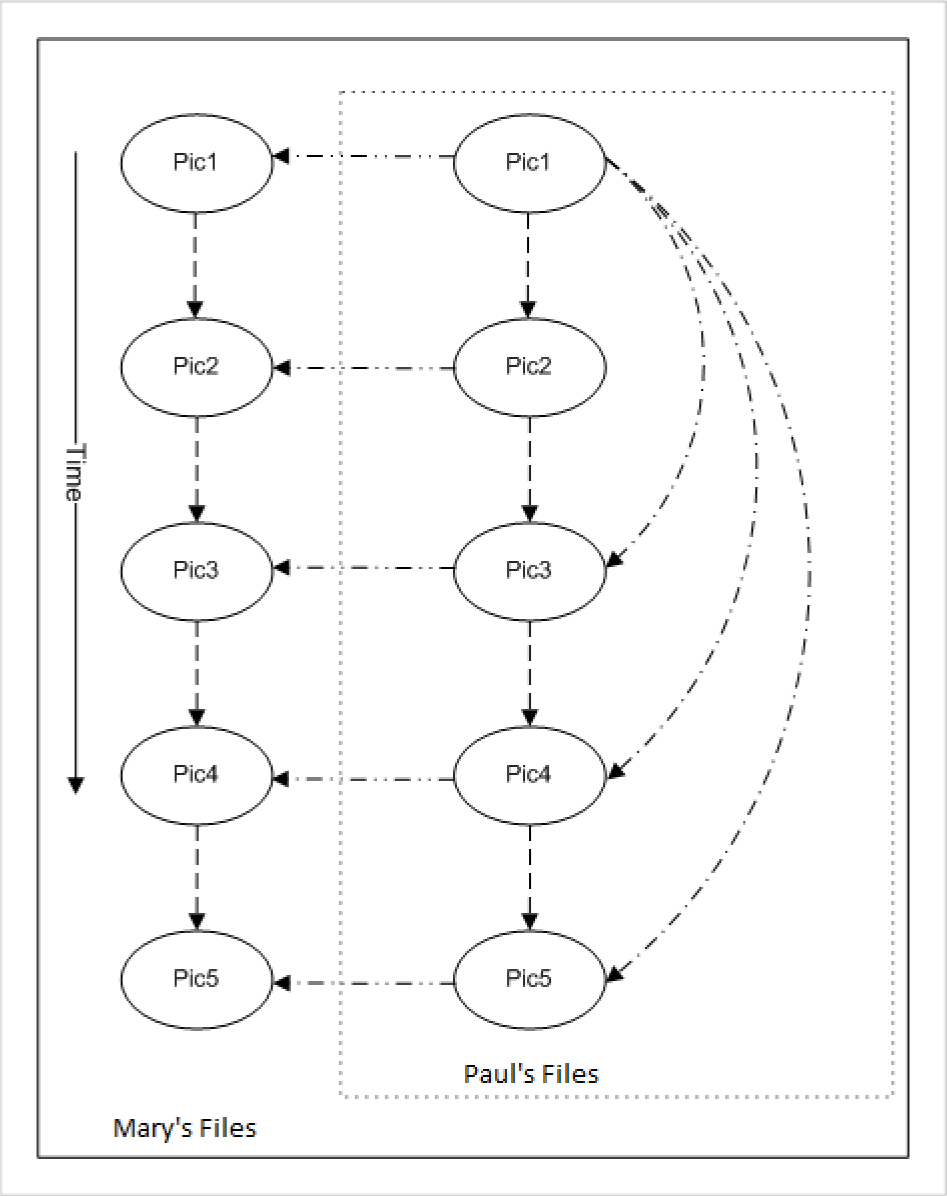


Figure 7.2: Case study re-visited

Due to this fact Mary found it very difficult to backtrack through the copies once a change was needed. During the collaboration Mary learns that changes till pic 5 was not actually needed and that an earlier version can be modified to suit the requirement and when Mary conveys the same to Paul, Paul would ask for the earlier version as he only has the master copy and the last copy which makes the collaboration difficult as it requires Mary to search for the different versions in the working directory.

The software that we built is able to track the side spawn-off, that is, when Mary receives a picture from Paul, she would copy and paste and then change the format of the picture to use in the paper. The whole process of copy and paste is captured by our system thus making it easy to search for the specific file. So when Mary would like to go the previous versions of the picture, she can select the picture from the working directory of the system and it will list all copy and paste relations thus making it simple to track the evolution.

Even though the system tracks the copy and paste process, we are yet notable to track the semantic or conceptual relationship between the files. One such scenario is when Mary requests a change in picture Paul does the change from the master file and not from the latest version, which is a semantic relationship, and the system that we built does not keep track of such conceptual relationships.

During the discussion about the software Peter pointed out to the system by the name Git. Git [GIT] is a distributed revision control system which is very fast and gives you maximum performance. It is able to monitor the various revisions to a certain document and is not based on a central server. The main difference between the Git and our software is the ability to capture the copy and paste there by giving more meaning to the original source.

7.7 Summary

In this chapter, we have discussed about the walkthrough on the software to the experts. There was individual walkthrough to each of the participating expert followed by a group walkthrough to the expert group. We also detailed out the outcome of each walkthrough. Furthermore we discussed about the changes we made to the software following the walkthrough.

8 Summary and future work

This thesis is about tracking the evolution in documents. In this chapter we will summarize the whole project and discuss on the future work. We will provide the findings of our research in a nutshell.

8.1 Summary

This project aimed at tracking the evolution that happens to a document over time. A scenario was looked into for the purpose of understanding the problem space; out of the scenario we developed the concept of Information and Data objects. The concepts of Information and Data objects were gradually developed with the requirements for the concepts.

We looked at a variety of related works and found that none of the present systems or related works exist that suits our requirement. A case study involving collaboration was looked into to understand the information flow to analyse and refine the concepts of Information and Data objects.

We designed a system with various components as proof of concept to track the evolution in a document. The system is explained with a detailed architecture followed by the implementation of the system. The implemented system was presented to experts who participated in the case study to get their feedback. An expert walkthrough was conducted to identify the problems in the system alone with the suggestion for further improvement.

We found that the implemented system is effectively able to track the evolution by monitoring the copy and paste actions of a user. It is also able to do this without any manual intervention which adds to the quality of the system. We also found that there still are some issues which need to be addressed for improving the quality of the system and the concept and is discussed in the future works.

In the implemented system, we were able to track the evolution information of a document. The evolution information is made available without any manual intervention by tracking the user copy and paste action through keyboard.

8.2 Future work

In this section we will discuss on the future research areas related to this paper. The future research areas could include, but not limited to, the following areas.

- Implementing the software as a cross-platform application – The system currently works on Windows OS and we would like the system to work on all operating systems so that a large user feedback can be collected.
- Incorporating the drag and drop, mouse copy and mouse paste actions to the application – The system currently tracks keyboard based copy and paste action and incorporating all possible methods would further improve the system.
- Making this application distributed so that evolution can be tracked more effectively – The system requires the application to be run locally and the database is not distributed. Making the application work on a distributed environment would definitely improve the evolution tracking capabilities of the system.
- More research to the concept so that it can track the semantic relationships in documents – keeping track of the semantic relationship is a very large research area and would require significant amount of time. Evolution can be tracked more efficiently if we are able to track the conceptual relations. M Rinck, a PhD student in the University of Waikato, is currently pursuing his research in this area.
- Realization of the concept of Information objects and Data objects – This paper proposes the use of new document concepts but still we were not able to realize the requirement that the Data object should be

independent of the Information object and this is an area of further research.

- Research on the feasibility into incorporating this concept at the operating system level – This paper discusses the implementation and the implementation is based on the resources that the OS provides. A further research into the feasibility of the concept to be introduced at the operating system level can reform the current document concepts.

Bibliography

[BV99] L Bendix and F Vitali. 1999. VTML for Fine-Grained Change Tracking in Editing Structured Documents. In *Proceedings of the 9th International Symposium on System Configuration Management* (SCM-9), Jacky Estublier (Ed.). Springer-Verlag, London, UK, 139-156.

[CL04] G Chin, Jr. and C S. Lansing. 2004. Capturing and supporting contexts for scientific data sharing via the biological sciences collaboratory. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (CSCW '04). ACM, New York, NY, USA, 409-418.

[CML] http://www.clipmagic.com/, Clip Magic online last accessed on May 2011

[CVS] http://www.nongnu.org/cvs/#documentation , The Open Source I online documentation last accessed on Nov 2010

[ECM] http://code.google.com/p/ecm/, The Open source Easy Clipboard Manager online code last accessed May 2011

[GIT] http://git-scm.com/documentation/ , The Fast version control system online documentation last accessed on August 2011.

[KM01] H Krottmaier, H Maurer. Transclusions in the 21st Century; Journal of Universal Computer Science 7, 12 (2001), 1125-1136.

[KM06] J Kolbitsch, H Maurer (June 2006). Transclusions in an HTML-Based Environment. Journal of Computing and Information Technology 14 (2): 161–174.

[LTX] http://www.latex-project.org/guides/ , The Latex online documentation last accessed on Sept 2010.

[RH11] M Rinck and A Hinze. 2011. Views on information objects: an exploratory user study. In *Proceedings of the 12th Annual Conference of the New Zealand Chapter of the ACM Special Interest Group on Computer-Human Interaction* (CHINZ '11). ACM, New York, NY, USA, 49-56.

[MP] http://msdn.microsoft.com/en-us/library/wtxbf3hh.aspx , The Microsoft Developer network ASP.NET Master Pages last accessed on Dec 2010

[MSWD] http://office.microsoft.com/en-us/word-help/ , The Microsoft word online documentation last accessed on Sept 2010.

[MSW] http://office.microsoft.com/en-us/word-help/CH010024383.aspx , The Microsoft Word Online Documentation on track changes and comments for MS Word last accessed on Dec 2010

[NELS95] T H Nelson. 1995. The heart of connection: hypermedia unified by transclusion. Commun. ACM 38, 8 (August 1995), 31-33.

[RKC08] C Ryu, H Kim, and H Cho. 2008. Reconstructing Evolution Process of Documents in Spatio-Temporal Analysis. In *Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology - Volume 01* (ICCIT '08), Vol. 1. IEEE Computer Society, Washington, DC, USA, 136-142.

[SGSG03] C A. N. Soules, G R. Goodson, J D. Strunk, and G R. Ganger. 2003. Metadata Efficiency in Versioning File Systems. In *Proceedings of the 2nd USENIX Conference on File and Storage Technologies* (FAST '03). USENIX Association, Berkeley, CA, USA, 43-58.

[SVN] http://svnbook.red-bean.com/ , The Subversion Version Control online documentation last accessed on Dec 2010

[TFS] http://msdn.microsoft.com/en-us/library/ms364061.aspx , The Microsoft Developer network on Team Foundation Server last accessed on Dec 2010

APPENDIX A – ETHICAL CONSENT FORM

Research Consent Form

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Ethics Committee, Faculty of Computing and Mathematical Sciences

Evolution of documents – Information and Data objects

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 walkthrough/ analysis of data, 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: ____

Additional Consent as Required

I agree / do not agree to my take notes during this expert walkthrough

Signed: ____

Name: ____

Date: ____

Researcher's Name and contact information:

Appu Mathew Jose

Department of Computer Science, The University of Waikato

Room G.2.06

appu.mat@gmail.com

Supervisor’s Name and contact information:

Dr. Annika Hinze

Department of Computer Science, The University of Waikato

Room G.2.04

hinze@cs.waikato.ac.nz

APPENDIX B – PARTICIPANT INFORMATION SHEET

Participant Information Sheet

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Ethics Committee, Faculty of Computing and Mathematical Sciences

Project Title

Evolution of documents – Information and Data objects

Purpose

This research is conducted as a part of my Master's research on evolution of documents – Information and Data objects. I am developing a software which is capable of tracking the evolution in documents and would like to conduct this expert walkthrough so as to gain more information on the problems of the system that I have developed and also to get suggestions which can be used to improve the quality of the system.

What is this research project about?

In my master's thesis I am developing a system which is able to track the evolution to a document over time by tracking the copy and paste actions of a user.

What will you have to do and how long will it take?

After you have signed the consent form, I will give you an expert walkthrough on the software. The expert walkthrough will consist of a presentation about the software that I have implemented. Finally we will conclude my taking your impressions, feedback and questions about the software. Altogether this will take about 20 minutes.

What will happen to the information collected?

The information collected will be used by the researcher to write parts of his Master's thesis. It is possible that articles and presentations may be the outcome of the research. Only the researcher and his supervisor will be privy to the notes. Afterwards, notes will be destroyed. 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 analysis has commenced on the data.
- 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:

Appu Mathew Jose

Department of Computer Science, The University of Waikato

Room G.2.06

appu.mat@gmail.com

Supervisor:

Dr. Annika Hinze

Department of Computer Science, The University of Waikato

Room G.2.04

hinze@cs.waikato.ac.nz