Peer Group Document and Citation Management

The peerDOC System

Technical Report #CS03-25-00

Siyabonga Mhlongo
University of Cape Town
Department of Computer Science
smhlongo@cs.uct.ac.za

Phathutshedzo Tshivhengwa
University of Cape Town
Department of Computer Science
ptshivhe@cs.uct.ac.za

Senate Mafike
University of Cape Town
Department of Computer Science
smafike@cs.uct.ac.za

-

Supervisor:
Dr. Hussein Suleman
University of Cape Town
Department of Computer Science

ABSTRACT

Researchers are usually overloaded with a lot of research papers and material that they have to efficiently organised to allow fast and efficient access. Over and above this, the process that these researchers have to go through to find the research material is often very tedious, sometimes involving many hours of searching on the World Wide Web.

This report discusses a case study that was done to investigate ways that can be used to build a system for assisting individuals to organise documents in an electronic "workbench". One of the key elements of this system is its ability to allow users to collaborate through simplifying the process of searching and downloading documents on a Local Area Network. By design this system would then eventually reduce external Internet traffic, and increase the efficiency of its users by eventually reducing the amount of time they spend on the World Wide Web locating research material.

The project was aimed at using the latest open standards as the basic building blocks for its framework. The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) was selected as the protocol that would be used for the collaboration aspect of the project.

OAI-PMH provides a low cost approach to interoperability. It is one of latest standards that are aimed at simplifying the dissemination of content.

This project was implemented in the form of components that operate independently. These components are then interfaced to form a bigger system.

This project was implemented using Java, to allowing easy portability between operating systems.

From the project it was discovered that the success of peer-to-peer downloading software relies heavily on popularity as users are the ones sharing the information. The time that was available for this project was not enough to fully explore and establish a large user base, thus experiments involving big volumes of users could not be conducted.

The project did however, result in the construction of a functional document manager, and also established a sound framework that could be further developed for future work.

Keywords

Document management, content dissemination, open standards, Open Archive Initiative (OAI)

1. INTRODUCTION

The increasing popularity of personal computers has been the key catalyst for the migration of many individuals from traditional filing cabinets to digital archives. This shift of paradigm has introduced a new problem of electronic document management. Traditional filing cabinets have tags and markings that individuals use for indexing the documents. Many researchers have been working on ways to organise digital documents using the indexing approach.

Digital systems have also afforded us new ways of collaborating and sharing information. The flexibility of digital documents allows individuals to efficiently share and distribute them. Although we realise these possibilities, people are still using either very primitive or nonstandard propri-

ety methods to organise and distribute documents. Efficient systems using open standards for disseminating and organising documents are still not popular.

We decided to call our project peerDOC.

2. BACKGROUND AND MOTIVATION

New knowledge is being created, almost everyday, by researchers through their research activities. According to Harnad [2], creating new knowledge is not enough. Harnad further elaborates, saying that even if the knowledge serves as an open-ended investment, it must be communicated to successor generations, but more importantly, to one's fellow researchers and peers, so they can apply, test and build upon it.

Stevan Harnad, a Professor of Cognitive Science at the University of Southampton, is a celebrated champion for open access to information. His philosophies include that which states that writers of scholarly research have the right and a responsibility to make their work as widely available as possible in order for the research to have the greatest impact on the further development of knowledge.

His main focus is on peer-reviewed journal article publications, and his research emphasises that most researchers write to publish their research, with minimum interest in the profit. Putting the articles online makes the knowledge available, but only to a limited audience, with obvious ramifications: how widely available is this information? Can the articles be reviewed by peers to a satisfying extent? How great an impact will this exposure have on the further development of knowledge?

To address these issues, the articles, including the pre-print and post peer-reviewed ones, should be available through archives (or repositories of scholarly knowledge), completely eliminating the cost associated with accessing these articles. This would also benefit other scholarly disciplines, within which research results are being produced at an increasingly rapid pace, for it would provide lower latency times than those experienced in the established journal system [3].

Since the early 1990s, there has been a number of these archives which do exactly what the above states, but as mentioned earlier on, new knowledge is being created at rapid rates and needs to be disseminated just as fast. To a certain extent, knowledge is being disseminated, but the rates thereof are not acceptable. Shearer, [4], labels these archives as isolated "islands" of information, which differ drastically in the ways in which they operate. In addition, although the knowledge may be available, scholars who may need this knowledge still have to hunt through these so-called "islands" for it, that is, there is no central place where a scholar can submit a query and have the system perform that query transparent to the scholar, searching through all the archives that are available and returning the results.

As a result of these domain boundary problems, the Open Archives Initiative (OAI) was born. Spearheaded by Herbert van de Sompel and Carl Lagoze of Cornell University, amongst others, the OAI aims to support the efficient dissemination of knowledge [3].

However, efficient dissemination of content (or knowledge) should be coupled, at both ends of the process, by adequate management thereof. For a researcher who aims to have their research work reach as far out to the intended audience as possible, it is essential that when that research work reaches a candidate it is dealt with accordingly. Going beyond research work, any knowledge that is propagable deserves to be managed in such a manner as to not introduce unnecessary management overhead to the parties involved.

Based on the latest standards in metadata transfer from the OAI to support federation, `peerDOC` aims at benefiting the researchers, and any users thereof, by providing a platform which supports efficient knowledge dissemination and management.

3. FRAMEWORK

This chapter discusses the general architecture of the project. The figure in Appendix 01 shows the overview of the system.

3.1 Overview

The system was inspired by the wish to make a document management system that uses the latest open standards for dissemination of data. The backbone of the project is the OAI-PMH (Open Archives Initiative - Protocol for Metadata Harvesting)which forms an integral part of the collaboration module in the project.

3.2 Architecture

The following picture shows the the architecture of the system.

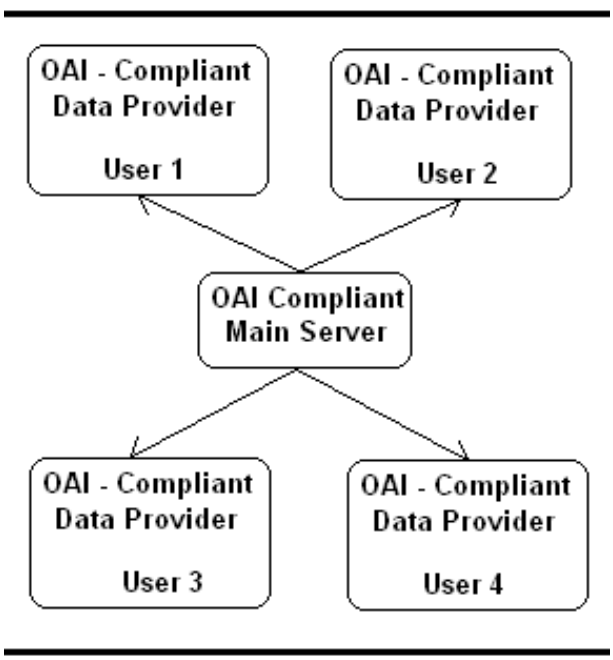

Figure 1: Diagram Showing the Architecture of the System

The diagram above shows the basic interconnection between users and the main server. The diagram in Appendix 01

shows a more detailed view of the both the user system and the server system. This diagram shows a break down of all the the components that make up the overall system.

All computers connected to the system are OAI compliant repositories which can be harvested by the harvester that is on the central server.

These are the components that constitute the server system.

- Repository (Central Database)
- Harvester
- Global Search Engine

These are the components that constitute the client system.

- Local Search Engine
- Document Organiser
- Local Metadata Repository
- File Transfer Tool
- Citation Generation

These components interface to build a system that is OAI compliant

4. IMPLEMENTATION

This section gives a brief discussion of how the project was implemented.

4.1 Document Management

As already mentioned in section 2, knowledge management is essential. This section gives an overview of what the *Document Organiser* is responsible for doing as well as how it is involved in the process of content dissemination.

4.1.1 The Document Organiser

The *Document Organiser* has been designed for the primary purpose of capturing metadata about documents from the researcher, while providing a simple workable interface through which this process takes place. Moreover, it serves as a tool for the researcher to use in managing the documents that the researcher may wish to manage, providing simple functions that are applicable to the documents that are under its management.

Figure 2 shows the interface of the *Document Organiser*. Another important role that the Document Organiser plays is serving as a front end for the process of content dissemination, for it is through the *Document Organiser* that the metadata is created. In later sections the process of getting the metadata from individual workstations (harvesting) is clarified.

The functions that the *Document Organiser* provide relate to the documents that are under its management as well as their corresponding metadata. The list below shows these functions:

Figure 2: The *Document Organisers'* Main Interface

- The metadata can be edited, providing numerous advantages.
- The user is offered the ability to search their local store of documents using a local search engine, based on the *Lucene*[1] infrastructure.
- The ability to generate citation entries in different predefined citation styles.
- Opening documents using appropriate applications that are found in that particular workstation.

4.2 Content Dissemination

Three main components were designed and implemented to cater for the dissemination of contents of the research material among the users registered with the system. The components referred above are the harvester, the downloader and the search engine. This section is a discussion of how the above-mentioned components were implemented.

4.2.1 The Harvester

"A harvester is a client application that issues OAI-PMH requests and is operated by a service provider as a means of harvesting metadata from a repository "[2]. The harvester was implemented based on the OAI-PMH protocol. The harvester was implemented to perform periodic and automatic harvests of the registered archives. The scheduling was achieved using a cron scheduler to schedule the harvester's execution for a specific instance of time.

The harvester was implemented to retrieve information about the location of the metadata from a registration database. Once the individual location addresses have been retrieved, the harvester sends OAI requests to each of the archives indicated by the addresses.

The harvest process is a selective process in which only certain metadata records are harvested. In this implementation the harvester uses selective harvesting based on the date on

[1]The *Lucene* package is an open source Java based project, offering searching capabilities.
[2]http://www.openarchives.org/

which a metadata record was created or modified. The harvester keeps a record of the date corresponding to the last harvest. This date is then used to mark the range of dates that the metadata must have been created or modified on, in order for it to be harvested. With this approach only the metadata that was created or modified after the last harvest date will be harvested.

There are a number of different ways in which the harvester could retrieve metadata records from the archivelets. One way is to harvest a single metadata record from each repository at a time, while the alternative way is to harvest a collection of metadata records from each archivelet at once. Each of the above methods is supported by the OAI-PMH protocol. For example, the GetRecord verb described in section 2 could be used to implement the first method, while the ListRecords verb is more suitable for the last method. The harvester was implemented to support the latter method, because of its simplicity and efficiency.

4.2.2 The Search Engine

The search engine was implemented using the inverted index search approach. This approach means that the search engine does not search through the input files, but searches an inverted index [5].

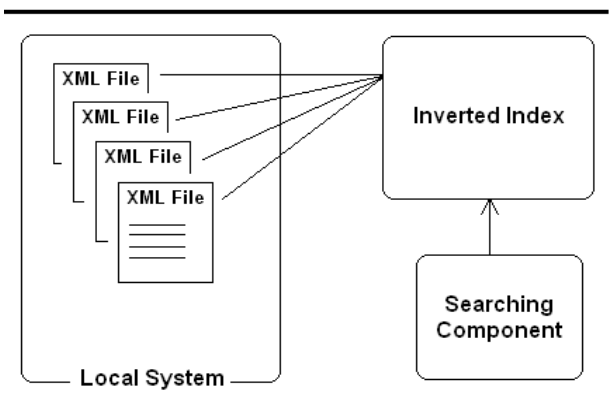

Figure 3: A Diagram Showing The Data Flow of the Search Engine

An inverted index is a sequence of (key, pointer) pairs where each pointer points to a file which contains the key value. The index is sorted on the key values to allow rapid searching for a particular key value using, e.g., a binary search. The index is "inverted" in the sense that the key value is used to find the record rather than the other way round.

The main challenge when building a search engine is building and maintaining this index. When search results are returned the search engines use some heuristics such as number of occurrences of words to determine the file that is more relevant to the search query and return the results in order of relevance.

The Lucene package which is an open source Java project was used for the creation and searching of the inverted index [1].

4.2.3 The Downloader

The peer downloader module of this project uses HTTP as its transport layer. This approach was derived from Gnuttella, which is one of the most popular peer to peer downloading protocols that also uses HTTP as a its transport layer. Taking this approach abstracts low level networking details, and reduces the amount of error checking that needs to be done at the transport layer of the OSI model.

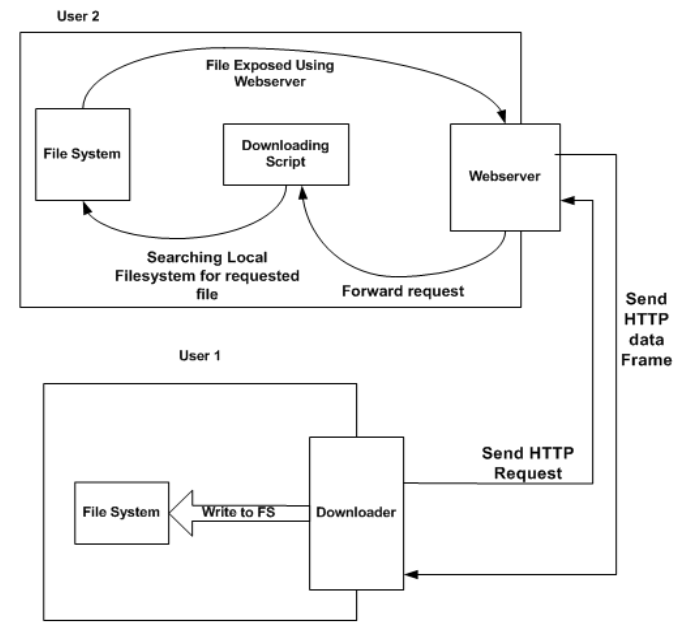

Figure 4: A Diagram Showing The Operation of the Downloader

The downloading user sends an HTTP request to the host computer requesting a file. The host computer has a script that locates the file and sends it back using HTTP. HTTP status codes are used to communicate in case errors occur.

5. CONCLUSIONS

The `peerDOC` system was aimed at:

- Investigating ways of developing a system that could non-intrusively manage user documents on a virtual work-bench.
- Investigating methods of increasing user efficiency when doing research by introducing ways to simplify the process of searching, downloading and storing the electronic research documents. *These methods included simplifying collaborating with peers and the generation of citations.*
- Investigating Open standard methodologies and technologies that exist and could be used to build a system that allows collaboration between users were also looked at.

Research also showed that there are already many components that exist that were built based on open standards

such as OAI-Protocol for Metadata Harvesting that can be put together to create a digital library, or a system that allows users to share information.

We found that for users to completely trust a system to manage their documents, the system needs to have an efficient and reliable searching module. Individuals are not usually too worried about the location of documents; their primary concern is usually around the efficiency of retrieval and access.

We also found that using open standards as a basis for a framework of a system allows the system to be easily accessible since the standards are well known.

6. FUTURE WORK

There are a few issues that due to time constraints and other external factors could not be addressed during the implementation of `peerDOC`.

6.1 Linking Central Servers

The system as it stands currently is designed to function efficiently on a Local Area Network (LAN) with a central server that stores metadata harvested from individual users. If there are several systems like this in one area, they could be linked up to form bigger networks.

The linking of these networks could be done by linking the servers, and allowing them to harvest metadata from each other. This would involve making the server repositories OAI compliant.

6.2 Self Archiving

One issue that we did not explore in great depth is how we can adapt the system, or extend the system framework to allow researchers to self archive their work internationally. This is one of the greater goals that initiatives like OAI and Budapest Open Access Initiative (BOAI) are aiming to eventually see happen. Currently the system allows efficient collaborations and information sharing between users on a local area networks we did not conduct tests to test the efficiency of the system for a (WAN) wide area network.

7. REFERENCES

[1] Brian Goetz. The Lucene search engine: Powerful, flexible, and free. [Available] http://www.javaworld.com/javaworld/jw-09-2000/jw-0915-lucene.html, Last Accessed: 24 September 2003.

[2] Stevan Harnad. Free at last: the future of peer-reviewed journals. D-Lib Magazine, 5(12), December 1999.

[3] Carl Lagoze and Herbert Van de Sompel. The open archives initiative: Building a low-barrier interoperability framework. In ACM/IEEE Joint Conference on Digital Libraries, pages 54–62, 2001.

[4] Kathleen Shearer. The open archives initiative - developing an interoperability framework for scholarly publishing. In CARL/ABRC Backgrounder, Series #5, 2002.

[5] Danny Sullivan. How Search Engines Work. [Available] http://searchenginewatch.com/webmasters/article.php/2168031, Last Accessed: 03 October 2003.