Honours Project Report

The **B**ushman **O**n**L**ine **D**ictionary Project

Archive Management System

Lebogang Molwantoa
lmolwantaa@cs.uct.ac.za

Supervised by: Dr Hussein Suleman
hussein@cs.uct.ac.za

Department of Computer Science
University of Cape Town

6 November 2009

Abstract

The preservation of culture and heritage is important, as the impact of Western influences deepens. It is estimated that in a few years, one of the oldest ethnic groups in the world - the Bushman people of South Africa - will pass on. The Bleek and Lloyd Collection is a set of documents that preserve the history and culture of the early Bushman inhabitants of the Western Cape of South Africa.

The **BOLD** (**B**ushman **O**n**L**ine **D**ictionary) project is an image-based dictionary that aims to integrate a recent set of digital scans into a dictionary that can be used as a live-reference for researchers and scholars alike.The system has 3 distinct components: An archive management system; a searching,browsing and display module; and an image-based translation module. The **BOLD** archive management system is the focus of this report, with the primary research aim of implementing an efficient and useful archive management system.

An iterative design model was chosen with regular prototyping for the implementation of the system. The archive management system was implemented using the open-source Fedora Commons digital repository for the back-end layer of the archive with Java Server Pages(JSP) technology as the Web-interface layer.

The archive management system was tested for both performance and usability. Performance testing was performed to test the efficiency of the back-end layer of the **BOLD** archive management system. The results of the performance test revealed the archive management system to be fairly efficient. The usability questionnaire revealed some flaws which hampered the usability of the system, but there is scope for improvement. This project met its aim of building an efficient and useful archive, with reasonable success.

Categories:

H.3.7 INFORMATION SYSTEMS[Information Storage and Retrieval]:Digital Libraries - Collection, Dissemination,Standards, User issues.
H.5.2 INFORMATION SYSTEMS[Information Interfaces and Presentation (eg. HCI)]:User Interfaces - User-Centered Design

Keywords: Digital Preservation, Fedora, Archive Management, Digital Repositories

Acknowledgements

First and foremost I would like to thank God, for granting me this opportunity to do this project and for walking with me and bringing me this far in the completion of my both my Honours academic year and the project.

A big note of gratitude to Dr. Hussein Suleman for his constant guidance and constructive criticism that helped greatly in the development of the project.

To my project partners - fellow *Bushmen* Kyle Williams and Sanvir Manilal - it was an honour to work with men like you with such a high level of work ethic. Thank you for the endless encouragement and support during trying times, when things were not working out. It was an amazing experience to work with you gentlemen.

To my wonderful girlfriend Vuyiswa - I thank you for your patience with me during this project, and the support you gave me when it was looking shaky. I appreciate you and all that you have done for me.

To the Computer Science Honours class of 2009 - it has been a year of many jokes, classic moments(both good and bad) and amazing friendships. You guys made this a memorable year for me.

And finally to my mother for her constant encouragement and for sacrificing everything for my education to get to where I am today, I dedicate this report to you. You are my hero.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cultural heritage in South Africa is in danger of being destroyed by degradation, inaccessibility and even natural disaster. There is thus a clear need to preserve cultural heritage and make it accessible for a long time.

A snapshot of South African heritage would be incomplete without mentioning the Bushmen people - one of the oldest known ethnic groups in the world. With the rapid influence of Western culture, there are now only a handful of these Bushmen people left in South Africa. It is estimated that in a few years the entire generation of Bushmen will have passed on [Suleman, 2007] and this highlights the need to preserve whatever ancient artefacts and knowledge that exists from the Bushmen people. Already as a consequence of Western influence, the ǀXam and the !Kun languages of the Bushmen people are extinct [Suleman, 2007], further motivating the need to preserve cultural heritage.

1.1 The BOLD Project

1.1.1 Motivation

The Bleek and Lloyd Collection [Skotnes, 2009] is a set of unique documents(narratives, drawings and documents) that preserve the history and culture of the early Bushman inhabitants of the Western Cape of South Africa - specifically the ǀXam and the !Kun people [CCA, 2009]. This set of documents has been scanned in and hyperlinked to provide access to researchers all over the world, to allow them to learn more about what is arguably one of the oldest known cultures [Skotnes, 2009].

The Centre for Curating the Archive (UCT Fine Arts) recently added to the existing collection a set of scans corresponding to a dictionary that can be used to interpret and understand the existing Bleek and Lloyd documents. A meaningful and innovative way of representing the dictionary images was sought, providing a platform for this project. An example of the scanned dictionary image is shown in Figure 1.1. This report presents the **BOLD** project - the **B**ushman **O**n**L**ine **D**ictionary project, that aims to integrate the collection of digital scans corresponding to a dictionary to the existing Bleek and Lloyd Collection, by implementing an image-based dictionary. This dictionary may be used to interpret and understand the original Bleek and Lloyd texts.

While there are numerous digital archive management systems in existence such as

Figure 1.1: Example of dictionary image

DSpace [DSpace.org, 2009] and Greenstone [Greenstone, 2009a], none of them tackle the problem of language and cultural preservation through the use of an image-based dictionary making this a unique project. The project also aims at providing a framework for future image-based dictionaries aimed at cultural preservation.

1.1.2 Proposed Solution

The **BOLD** project - is an image-based dictionary that aims to integrate the recent set of digital scans into a dictionary that can be used as a live-reference for researchers and scholars alike.

The proposed system that implements the dictionary is split into 3 distinct components that are integrated. The first component involved the creation and the management of a digital archive to store and manage the scanned images; the second component is a searching, browsing and user interface component; and the third component is the interaction between the existing Bleek and Lloyd collections with the dictionaries, using image-based translation.

- Archive Management System
  The archive management system can be considered as the back end to the system. The archive system is a repository for the set of dictionary images as well as their associated metadata. This component of the project was designed and implemented by the author, and is the focus of this report.

- Searching, Browsing and Display
  This portion of the project allows the user to view the dictionary in various ways,namely: A thumbnail list view of the images; a linear textual list of the words that can be scrolled; and finally hyperlinks that link the words to existing material from the Bleek and Lloyd collection. This component was designed and implemented by Sanvir Manilal.Details of the Searching, Browsing and Display module can be found in Manilal's report [Manilal, 2009].

- Image-based translation

This component of the project involves providing for interaction between the existing Bleek and Lloyd collections and the newly scanned dictionaries, by making use of image matching techniques to match words in the collection of scanned notebooks to words in the dictionaries. Users select a word in the scanned notebooks which will be used as the search key for content based image retrieval (CBIR). Using this selected key word, the dictionary will be searched for the same word and the user will be provided with the scanned page with the translation of the word. This component was designed and implemented by Kyle Williams. Details of the translation module can be found in Williams' report [Williams, 2009].

Figure 1.2 shows a high-level overview of the proposed system - displaying how the proposed system connects the different components.

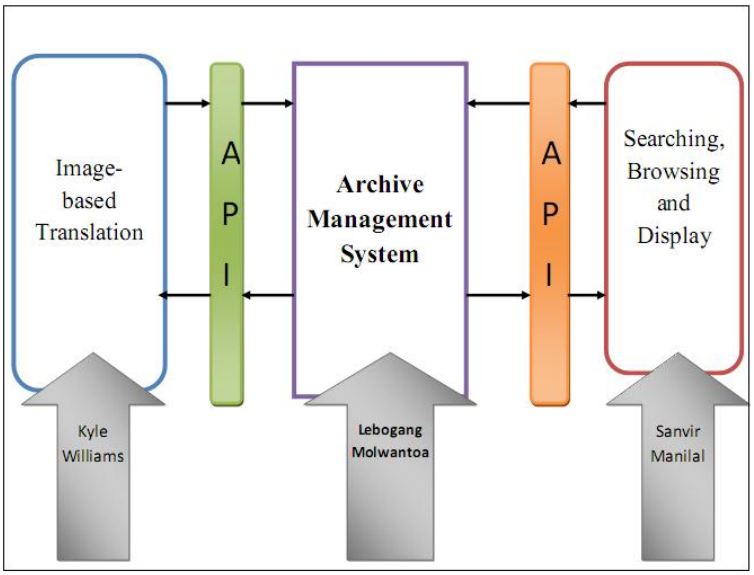

Figure 1.2: High-level view of proposed system

1.2 The BOLD Archive Management System

1.2.1 Problem Statement

The addition of the recent set of digital scans to the existing collection, presented a new challenge in managing the Bleek and Lloyd collection . A meaningful and efficient way was needed to preserve these images, thus providing the framework for this project. There are existing digital preservation systems for the management and preservation of cultural heritage and language. However there are no digital archival systems that are built specifically with the aim of providing a framework for archiving large collections that form an image-based dictionary. Existing digital repository systems such as Greenstone, DSpace and EPrints are not known for being able to archive heritage artefacts, and present them in a meaningful manner. This provided the platform for the **BOLD** project's archive management system.

1.2.2 Motivation

Physical degradation and the decay of storage media are two of the strongest motivators for the need to build an archive management system for the dictionary images. The **BOLD** archive management system, attempts to solve both of these issues while simultaneously providing ease of access to the scanned dictionary images.

1.2.3 Aim of Research

This project will address the issue of how to preserve cultural heritage when challenged with technological obsolescence. The research question posed in this project is: *Is it possible to create a reusable, generic archival system that allows users to access an image-based dictionary?*

In addressing the archive management system of this project, another research question is posed namely: *Can we develop a useful and efficient archival system?*

1.3 Report Outline

This report presents the work done in the implementation of the archive management system of the **BOLD** project. Chapter 2 discusses the background to the project by reviewing literature on related work as well as discussing the technologies that are used in digital preservation and archiving. Chapter 3 presents the design and the implementation of the BOLD project, and evaluation of the research hypotheses as well as the system is done in Chapter 4. The report is concluded in Chapter 5 and possible future works and extensions to the project are presented in Chapter 6.

Chapter 2

Background

2.1 Introduction

South Africa is a country that is rich in culture and heritage. It is thus important to preserve the culture and artefacts associated with South African heritage, possibly in a digital format. Digital preservation must also consider issues of accessibility [Moore and Marciano, 2005] - that is, ensuring that the artefacts are stored in such a manner that they can be accessed easily by a novice and still provide an educational framework for people wanting to learn more about South Africa's history.

Preserving cultural heritage is more than just a technical process of perpetuating digital signals over long periods of time [Lavoie and Dempsey, 2004]. It is also a social and cultural process, in the sense of deciding what materials should be preserved, and in what form. It is often described as an *economic process* [Lavoie and Dempsey, 2004], in the sense of matching limited resources with ambitious objectives. And perhaps most importantly, it is a continuous process that must be sustained to preserve cultural heritage.

This chapter gives some background relating to the Bleek and Lloyd Collection, followed by a discussion of digital preservation and related work pertaining to the preservation of cultural heritage. Finally, software technologies that are employed in addressing the issue of digital preservation are discussed.

2.2 The Bleek and Lloyd Collection

The Bleek and Lloyd Collection is presently archived at the University of Cape Town, the National Library as well as the Iziko South African Museum [Aluka, 2009]. The collection came about as a result of the collective pioneering efforts of two colonial scholars - Wilhelm Bleek and Lucy Lloyd who set about to document the language and culture of the ǀXam and !Kun people, in the face of cultural extinction [CCA, 2009]. The narratives documented by Bleek and Lloyd resulted in more than 13 000 pages, describing the culture and language of the ǀXam and !Kun people. Today the ǀXam language is no longer spoken by a single person and the pages and images reproduced by the Centre for Curating the Archive [CCA, 2009], are almost all that remain of the language and ideas of the ǀXam and !Kun people. The collection is recognised as a national treasure and has recently been entered into UNESCO's Memory of the World Register [CCA, 2009].

In *Digital Libraries Without Databases: The Bleek and Lloyd Collection* [Suleman, 2007], Suleman devised an XML-centric approach to manage the Bleek and Lloyd Collection, showing the XML-centric approach to be more efficient than the traditional database model . The resulting work by Suleman is available in DVD format, that accompanies a book by the title *Claim to the Country* [CCA, 2009], [Skotnes, 2009]. One issue addressed by Suleman in his paper was the issue of scalability, as XML-based solutions (while being reasonably scalable), may not have have enough capacity to deal with arbitrary large amounts of information [Suleman, 2007].

2.3 Digital Preservation

2.3.1 Definition

Digital preservation can be defined as the set of processes and activities that ensure continued access to information [Moore and Marciano, 2005] - all kinds of records, scientific and cultural heritage - existing in digital formats. The preservation of digital entities requires data management technologies that are provided by digital libraries and data grids. Digital archives are dedicated to the long-term preservation of data with the directive to ensure that they capture and preserve the data in a manner such that it can be accessed and presented at any time [Ludäscher et al., 2001].

2.3.2 Motivation for digital preservation

The preservation of data in a digital format is important in order to make content easily and readily available. Digital archives provide preservation environments that assure the authenticity and integrity of digital entities [Zorich, 2003].

2.3.3 Strategies deployed in digital preservation

The Online Computer Library provided a 4-pronged approach in addressing the need to digitally preserve data [OCLC, 2009]. Their approach was:

- Assessing the risk presented by technology.
- Providing access to the digital content.
- Determining and attaching the appropriate metadata to the digital content.
- Determining what type of digital format should be applied.

Other strategies that are employed when preserving data are [OCLC, 2009]:

1. Refreshing - the transfer of data between two types of storage medium.
2. Migration - the transferring of data to a newer system environment.
3. Replication - the creation of duplicate copies of data on one or more systems.
4. Emulation -the replication of the functionality of an obsolete system.

2.3.4 Digital preservation standards

The Open Archival Information System (OAIS) model provides an architecture for conducting digital preservation research and experimentation [Ray et al., 2002]. The OAIS model consists of an organisation of people and systems whose mission is to ensure that information is preserved and is accessible to the community [OAIS, 2009].

2.3.5 Language and cultural preservation

The preservation of cultural artefacts and digital documents forms a great portion of what is entailed in information preservation. A lot of research has been conducted on methods of preserving cultural heritage. Such initiatives are discussed in the following section.

2.4 Related Work

Cultural heritage in many areas around the world is endangered, mainly due to the overwhelming influence of Western civilisation, ideals and lifestyles [Liu and Tseng, 2004]. A problem in a lot of cultures, especially in Africa, is that cultural heritage is not preserved and is in danger of being destroyed by degradation, inaccessibility or even natural disasters. Thus, there is a clear need to digitise and archive these cultural artefacts.

The CAMA (Contemporary African Music and Art Archive) is an archive that aims to digitally capture as much of contemporary African culture as possible [Marsden et al., 2002]. This is done through the usage of camcorders, digital cameras and audio recorders. The CAMA project also aims to ensure that the archive is accessible to everyone as well as building a system which can present African art in a meaningful way [Marsden et al., 2002].

Many museums and libraries digitise their collections of historical artefacts to preserve them and also to make them accessible. A good example of this is the Armarius archive, which is an online document management system for ancient manuscripts [Doumat et al., 2008]. The Armarius archive digitises historical documents in a dynamic archive that can be accessed by anyone. The Armarius archive digitises historical documents by storing them in a database, structuring these documents as well as providing a platform to access the collection. Some of the collections that are found in the archive are the Arabic ancient manuscripts found in Timbuktu, manuscripts from mathematicians in the 14th century as well as Syrian manuscripts. In addition, the archive uses an online annotation service for researchers and scholars.

The Travellers in the Middle East Archive (TIMEA) [Spiro et al., 2006] is a digital archive that enables users to understand the explorations in the Middle East in the period between the 18th and 20th centuries. The TIMEA archive aims at enabling wide access to cultural heritage material while simultaneously promoting research skills amongst users of the archives - who are mainly historians and scholars. TIMEA is currently providing access to a growing collection of images and pages of encoded text [Spiro et al., 2006]. The archive integrates already existing technologies i.e. GIS maps, digital asset management software called DSpace [DSpace.org, 2009] for texts and images, as well as

Connexions which encompasses contextual research and teaching material.

The Greek Orthodox Archdiocese of America (GOA) has a rich and varied collection of important artefacts that are in the form of historical iconography, art, letters and memorabilia [Nicolakis et al., 2003]. Many of these artefacts are in a fragile state and cannot be handled by the many history scholars who wish to study them - an example is a lot of the church letters are written on very fragile onion skin paper. As a result, through the Department of Internet Ministries, the GOA has undertaken the project of digitising these artefacts with the main purpose of making them readily available for appropriate purposes. The artefacts are used mainly by theology scholars and historians who are interested in studying these artefacts.

There are many other initiatives that preserve languages and cultures. The Canadian Heritage Information Network's (CHIN) Virtual Museum is an online digital library that collects the contents of Canadian Museums and makes it available for the public to use [ECHO, 2009]. North Carolina's Exploring Cultural Heritage Online (ECHO) promotes the use of digital technologies in order to broaden and enhance access to the cultural heritage of the state of North Carolina as well as to encourage collaboration between all other states' cultural resource institutions [ECHO, 2009]. ECHO is an online portal to other online special collections of North Carolina's archives, museums and libraries .

The many archives in existence that preserve cultural artefacts further justify the need to preserve these artefacts as a means to ensure that they last for future generations. All of the above mentioned systems use archive management software and allow users to readily access the collection of digital cultural artefacts. Some systems such as TIMEA allow users to engage thoroughly with the ancient manuscripts through the integration of GIS(Geographical Information Systems) maps and contextual material. However, none of these systems integrate a dictionary that can be used as a live reference by researchers and other people who access the archives.

It is on this very basis that the idea was formed of integrating an online dictionary as a live reference for scholars who access the Bleek and Lloyd Collection. The integrating of this dictionary will be done through a digital archive and in the next section a discussion of related software technologies is presented.

2.5 Software Technologies

2.5.1 Digital Repository Systems

Hardware and software technologies evolve more rapidly than physical media decay, and this is one of the major challenges faced by archivists of digital information - a phenomenon referred to as technological obsolescence of the infrastructure that is used to access and present the information that is archived [Ludäscher et al., 2001].

A digital repository system is software that is used to build a digital archive and provide services that help to manage and organise the repository. There are different forms of digital repository systems to manage the wide variety of digital objects in a way that is most suited to that object.

2.5.2 DSpace

DSpace is an open-source, cross-platform software package, written in Java that provides tools for the management of digital assets [Baudoin and Branschofsky, 2003]. DSpace preserves and enables easy and open access to all types of digital content including text, images, moving images and data sets [Baudoin and Branschofsky, 2003]. There are numerous benefits to using DSpace. These include:

- Providing a platform for long term storage of digital material.
- Reaching a wider audience through exposure to search engines such as Google.
- Having a persistent network identifier for your work that never changes or breaks.

DSpace is implemented in Java and JSP, using the Java Servlet API. It also supports the use of relational databases such as Oracle and PostgreSQL, as well as ApacheHTTPD for certificate support. It supports the Open Archives Initiative Protocol for Metadata Harvesting, which is a protocol developed by the Open Archives Initiative (OAI) [Initiative, 2009] to collect the metadata descriptions of the records in an archive so that services can be built using metadata from many archives [Initiative, 2009]. The latest stable release of DSpace is version 1.5.2.

2.5.3 Greenstone

Greenstone is a suite of software for building and distributing digital library collections. It provides a way of organising information and publishing it on the Internet or on CD-ROM [Greenstone, 2009a]. According to the Greenstone Digital Library Software website, the aim of the software is to empower users particularly in universities, libraries and other public service institutions to build their own digital libraries. Greenstone is easy to use and the usage of the system is made easier through the Greenstone Librarian Interface (GLI) [Greenstone, 2009b], which is shown in Figure 2.1. Greenstone is capable of building up multi-media digital documents such as text, PDF, audio and video [Greenstone, 2009b]. The text, PDF, HTML and similar documents are converted into Greenstone Archive Format (GAF) which is an XML equivalent format. A problem

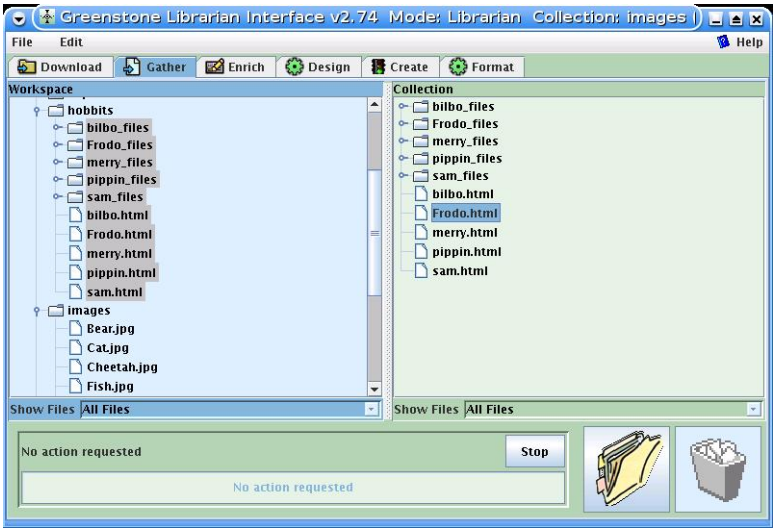

Figure 2.1: Greenstone Librarian Interface(*Image licensed under GNU GPL Version 3*)

which was highlighted in a paper on the creation of an archive for the Bleek and Lloyd collection [Suleman, 2007] is that Greenstone may present a portability problem as it requires some basic software installation, and this may become an issue when the system is meant to work on any arbitrary system.

2.5.4 Fedora

The Flexible Extensible Digital Object and Repository Architecture (Fedora) is a general-purpose, open-source digital object repository system Payette S. [1998]. Fedora is an open source system for the collection and management of different types of digital objects [Payette S., 1998]. Fedora is built on the principle that the best way of integrating data and interfaces - as distinct modules - is by using the principles of interoperability and extensibility. Fedora is not a complete application with all indexing, querying and discovery applications of a digital repository; it is merely a framework upon which other systems may be built. Fedora provides a general purpose management layer for the management of digital objects. The key features of the Fedora system are:

- The support of heterogeneous data types and being able to adapt to new data types.
- The ability to specify multiple content disseminations of digital objects.
- Associating rights management schemes with these disseminations.

Fedora uses a compound digital object design which aggregates one or more content items into the same digital object. Content items can be of any format and can either be stored locally in the repository, or stored externally and just referenced by the digital object. The Fedora digital object model is simple and flexible so that many different kinds of digital objects can be created, yet the generic nature of the Fedora model allows all objects to be managed in a consistent manner in a Fedora repository [Fedora, 2009]. The basic components of a Fedora object, as shown in Figure 2.2 are:

- Object PID - A unique persistent identifier for the object.
- Object Properties - Consists of a set of Fedora-defined properties that describe the object. This includes the metadata that describes the object.
- Datastream(s) - Represents the actual content of the stored object. A datastream is the element in the Fedora digital object that represents a content item. An object can have multiple datastreams, as shown in the Figure 2.2.

The Fedora architecture is divided into four subsystems and a Web Services layer. The core subsystem layer consists of the management subsystem which manages all the operations on the digital objects and an access subsystem that implements the operations that are necessary for disseminating objects and discovering more information and behaviours for an object. Fedora allows for the interchange between Fedora and XML-based applications and this mechanism facilitates archiving. Fedora supports the import and export of digital objects in a variety of XML formats [Fedora, 2009].

2.5.5 Other digital library technologies

Relational database model

The relational database has a naturally close relationship with many Digital Library Systems such as DSpace and Greenstone, which by default use mySQL and Postgres

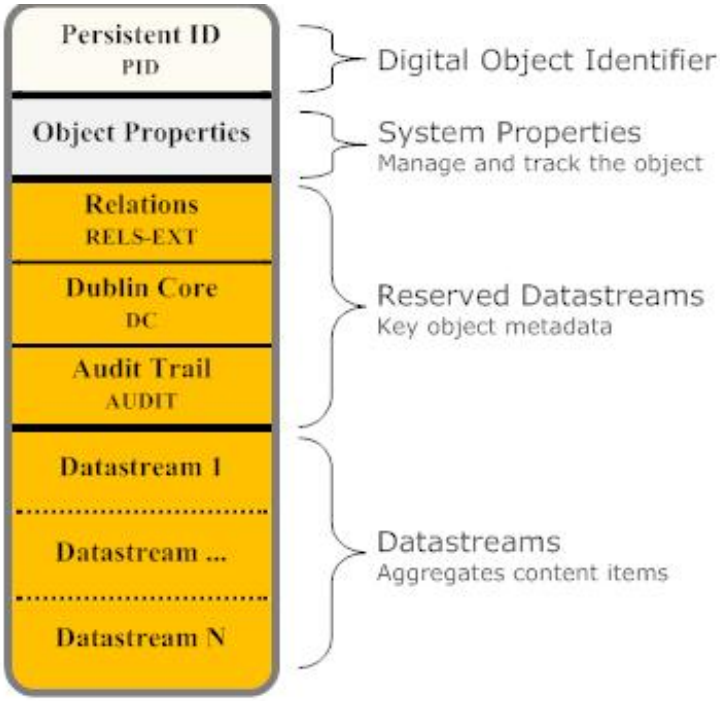

Figure 2.2: Fedora Object Model

database systems respectively [Baudoin and Branschofsky, 2003] , [Greenstone, 2009a] to hold their primary metadata repositories. The relational database model provides a useful platform for a basic mechanism which enables insertion, removal and updating of an archive. A well established query structure as well as efficient existing operations on the database systems provides a good argument in favour of the use of database systems for metadata storage in digital archives.

But databases also present another range of problems. In a paper on *Digital Libraries Without Databases: The Bleek and Lloyd Dictionaries* [Suleman, 2007], the author summarises some of the problems present in a database archive system, versus an XML-centric approach. A problem with a database system is that of the database not being platform-independent. This can be a problem on an arbitrary system, where the data needs to be extracted before it can be processed. In addition there is often a need for an administrator to run the database, which usually means a heavy reliance on this person.

XML-Based archival solutions

An XML-based archival infrastructure complies with the notion of not requiring special access software and is open and simple to use for both humans and programs(via parsing). Because of user-defined tags and the fact that some documents contain some schema information in the structure of their parse trees, XML can be viewed as a generic and self-describing data format [Natu and Mendonca, 2003]. XML-centric solutions have been recommended in the preservation of heritage-based digital collections because of their expected long term method of preserving data [Suleman, 2007]. The big concerns with using XML-based digital archives are the issues of scalability as well as the customisation of interfaces to these archives - allowing the user of a heritage-based digital archive to customise their interface to facilitate browsing,searching, upload and deletion

of objects.

Java Content Repository(JCR) API

The Content Repository API for Java (JCR) is a specification for the Java platform that allows access to contents in a repository in a standard manner [Barik, 2009]. The JCR is defined as an object database with searching, storage and retrieval features. JCR can be found in content management systems as well as in the storage of metadata.

The data in the JCR is stored in a tree data structure consisting of Nodes with associated properties. Data is stored in the properties which hold arbitrary length binary data and strings. Queries in JCR are performed using XPath and it also has the ability to support some standard form of SQL.

2.6 Summary

This chapter has shown the available archive systems that are used to preserve cultural artefacts, as well as the various initiatives that address the issue of cultural preservation. Based on the findings in the literature review, Fedora was chosen for the implementation of the Bushman OnLine Dictionary digital archive.

Chapter 3

Design and Implementation

3.1 Design Motivation

This chapter presents the design of the archive management system of the **BOLD** project. The approach that was chosen was iterative in nature and focused largely on user-centered design(UCD). The functionality of the archive was developed in accordance with user requirements, and thus the system was developed using an iterative design model with rapid prototyping. Users where involved in the requirements gathering and the final prototype sessions.

A brief overview of the system is given and the technology tools used in the development are presented. Finally a discussion of the requirements gathering for the development, and implementation of the archive management system - using the Fedora Commons digital repository API - is discussed.

3.2 System Overview

The proposed system that implemented the Bushman OnLine Dictionary project is split into 3 components. The first component will involve the creation and the management of a digital archive; the second component is a searching and browsing facility; and the third component is the interaction between the existing Bleek and Lloyd collections with the dictionaries, using an image-based translation module. The components of the dictionary where split vertically - as shown in Figure 3.1, which gives a high-level view of the system archive system and its connections to the other modules of the **BOLD** project. The repository's main functions are firstly to store and to manage the images and metadata stored in the archive. The second function is that of providing some form of application programmer interface (API) in order to facilitate the searching and browsing facilities that are to be provided in the dictionary.

3.3 Software Technologies and Tools employed

Figure 3.2 gives an overview of the archive management system - which is the focus of this report. The archive management system consists of a back-end layer built on top of the Fedora Commons Digital repository, as well as a Web-interface layer for the user-interface. A brief description of the technologies and tools employed is given below:

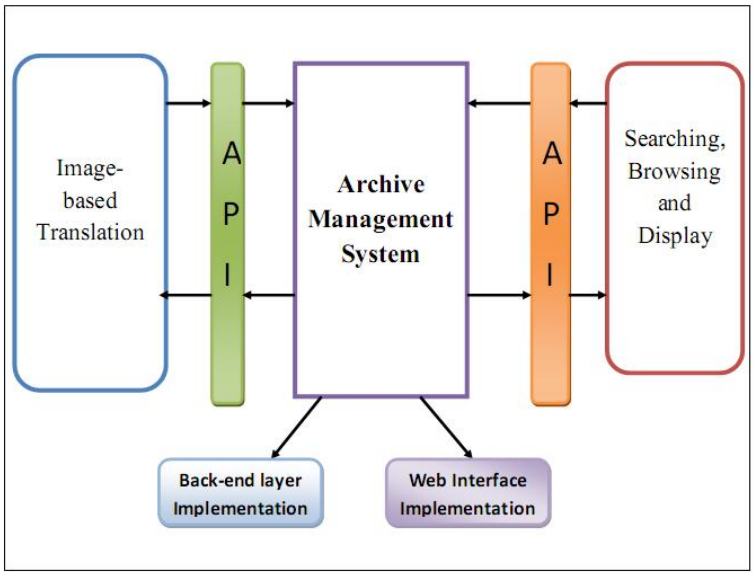

Figure 3.1: High-level view of proposed system - with API connections

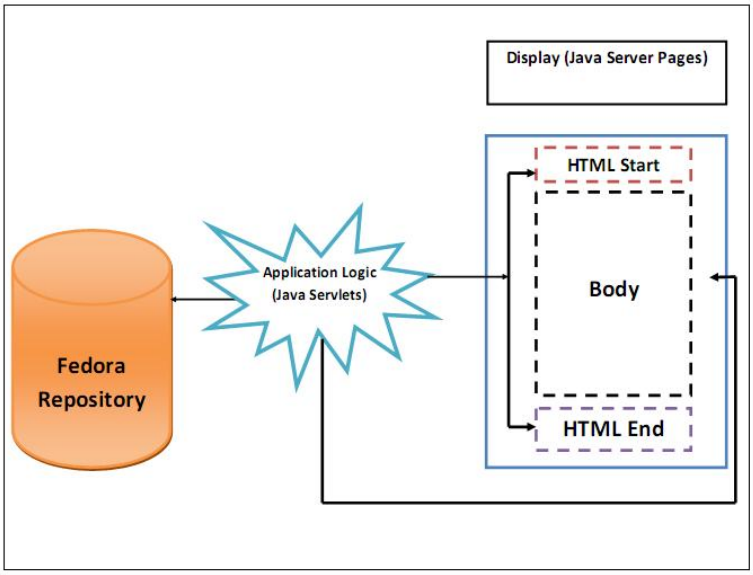

Figure 3.2: Archive Collection Management System architecture overview

3.3.1 Backend

Flexible and Extensible Digital Object and Repository Architecture (FEDORA) [Payette S., 1998] is the digital repository system that was chosen to build the archive, to store the dictionary images. Fedora is a general-purpose, open-source digital object repository system [Fedora, 2009] for the storage, management and dissemination of different types of digital objects and their relationships. Some of the features of the Fedora Commons digital repository include:

- The storage of all types of content as well as its metadata.
- Scalability to a million objects.

- The ability to customise any front-end to it.

The latest version of the Fedora Commons digital repository was installed (version 3.2.1) using MySQL server 5.1 as the environment database system that Fedora runs on. Development of the backend was done using the Fedora Access and Management API(API-A and API-M). Fedora API-A defines a client-side interface to enable repository managers to access the objects stored in the archive, while the Fedora API-M allows the repository manager to manage the repository(i.e creating, modifying and deleting digital objects, or parts within digital objects) [Fedora, 2009].

3.3.2 Development Environment

The BOLD project was developed using the Java framework for developing Web applications. The development of the archive management system was done in Windows XP, using Netbeans 6.5.1 Integrated Development Environment(IDE), and the Web application was deployed on an Apache Tomcat server that was bundled with the Fedora Commons Digital repository installer. Development and testing was performed on a personal Packard Bell IMedia 5225 desktop computer as well as a laboratory machine in the Computer Science Honours laboratory.

3.3.3 Web Interface Development

Java Server Pages (JSP) technology was used for front-end display of the contents of the repository via the Web. JSP provided a simplified way of creating dynamic Web content to allow for communication with the Fedora repository. Java Server Pages(JSP) was complemented by the Apache Struts framework [Apache, 2009].

All software technologies and tools used in development are open source, and the project source is released under GNU Public License version 3, as mentioned in the project proposal.

3.4 Requirements Gathering

The major requirement of the BOLD project is to integrate the dictionary into the main Bleek and Lloyd Collection so that it can be used as a live reference by researchers so that its contents can be preserved for future use. Development of the dictionary was done using an iterative design model. The iterations are discussed below:

3.4.1 Iteration 1 - Brainstorming session and initial system design

The preliminary meeting was a brainstorming session to ask questions and to move forward towards developing a solution for creating the **BOLD** visual-dictionary. With the guidance of the supervisor, the following aspects of the system were addressed, namely:

- The visual dictionary needs to have some form of archive with basic searching and browsing functionality.
- Through the archive, there must be a way to browse through it in different ways, and also to have different views of the information represented in the archives.

- Integrate the archive with the existing text in the Bleek and Lloyd Collection.

From these issues, the primary features of the framework for creating the **BOLD** archive mangement system were agreed upon, namely:

1. An archive portion that stores the images and their associated metadata.
2. Searching functionality for the archive.
3. A way of displaying the archive.

A design meeting was held with members of the UCT Fine Arts Archive, to refine the initial system design and extract more requirements of what the system needs to do. The session also gave the developers the opportunity to ask questions relating to the size and nature of the collection, the associated metadata for the files and what functions they would like to see implemented both for the entire system, as well as the archive management system. A copy of the original scanned TIFF dictionary images was stored on a local server in the Computer Science Department.

3.4.2 Iteration 2 - Initial Feasibility Demonstration

The second iteration was a proof of concept demonstration to show that the chosen technology was feasible for the project. This iteration also ensured that the core difficult portion of the project was addressed. For this iteration, the focus was on installing the Fedora Commons digital repository software and utilising the functionality of the API by developing features that can perform basic archive management operations.

A Simple Object Access Protocol (SOAP) client was implemented using the Fedora repository API. Basic operations such as ingesting and deleting of objects were completed. The feasibility demonstration affirmed the choice of Fedora as the underlying digital repository for the preservation of the dictionary entries.

3.4.3 Iteration 3 - Version 2 Prototype

The initial system was developed based on recommendations from users (UCT Fine Arts staff members Thomas Cartwright and Cara van der Westhuizen) as well as members of the Computer Science Department Digital Libraries group.

Paper-based prototype

Following on from the initial feasibility demonstration, a low-fidelity paper prototype was produced and evaluated. Figure 3.3 shows a very basic preliminary low-fidelity prototype of the system. The initial paper-based prototype was evaluated and critiqued to allow for further improvements in the design.The design was evaluated amongst the developers,project supervisor as well as some users - mainly fellow students in the Computer Science honours class with experience in Web development and information management. From the evaluation, it was established that an administrator that manages the archive should be able to perform the following functions:

1. Uploading of images
   The archive administrator must be able to add new images to the dictionary collection from a specified directory. Batch ingest using zip utilities was also pipelined

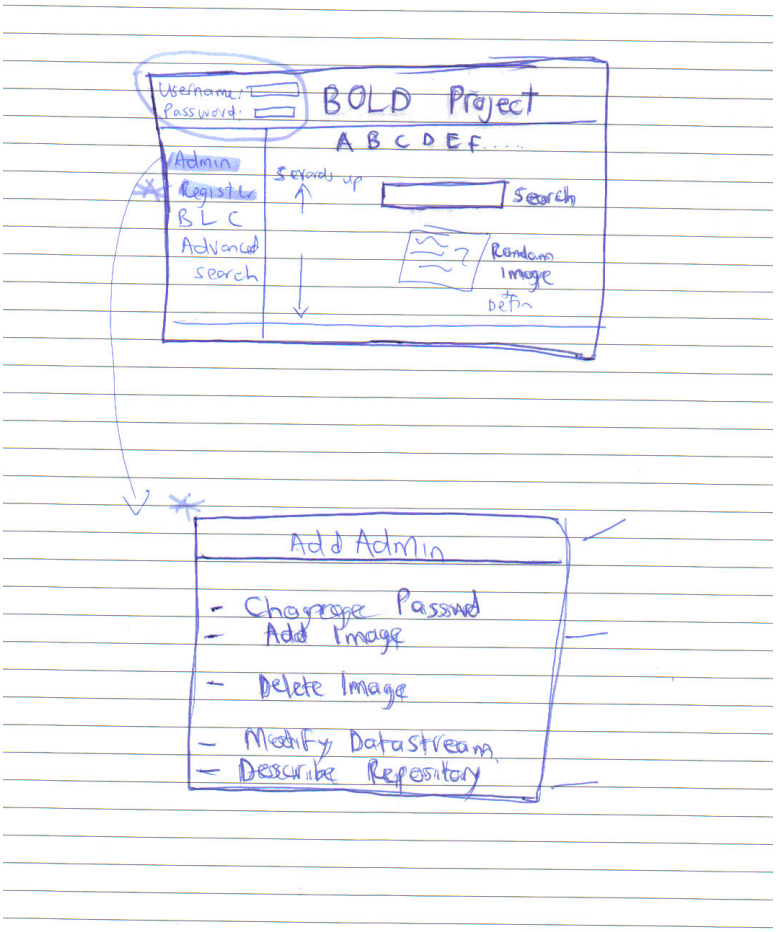

Figure 3.3: Initial paper-based prototype

as an additional upload feature but was omitted due to time constraints. In addition, the administrator should be able to add metadata about the object during the upload of images to the repository.

2. Deleting images
   The archive administration Web application should support the ability to delete multiple or single images from the archive. Ideally a listing of all the items should be displayed and the archive administrator should have the ability to select the respective item(s) that he/she wants to delete from the archive.

3. Metadata editing
   The archive administration component of the visual based dictionary should also allow the archive administrator to modify the metadata of the object.

Version 2 prototype

Development of the prototype was done in accordance with the functionality discussed above. The prototype was displayed to members of the Computer Science Digital Libraries Laboratory and members of the UCT Fine Arts department, with the purpose of refining the design of the archive management system. Figure 3.4 displays the login page of the system. The first criticism leveled against the prototype was the interface. It

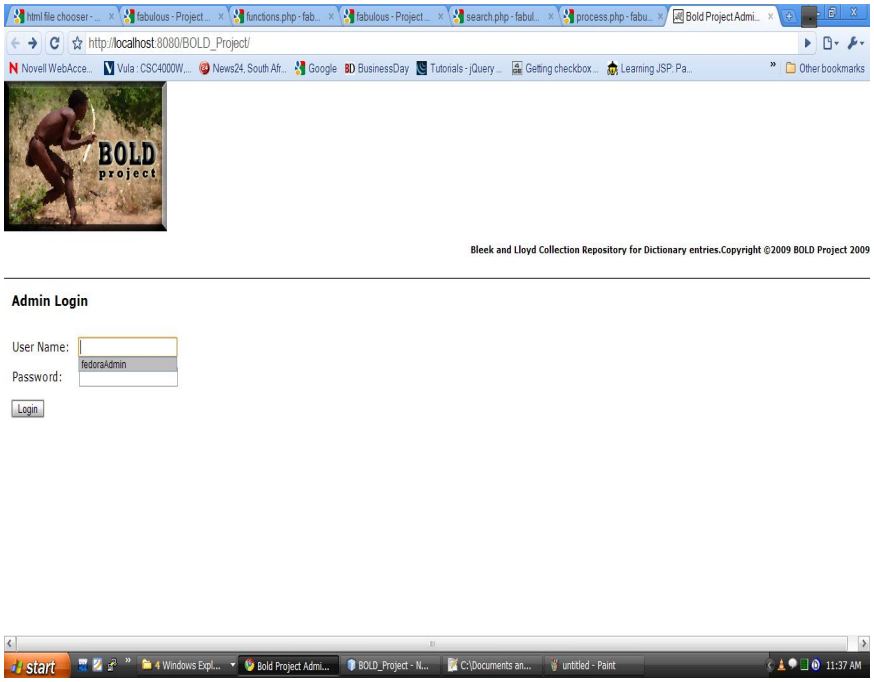

Figure 3.4: Version 2- Login Page

was suggested that the interface be made to look *'nicer'*, as the white background was said to be too plain. Figure 3.5 displays the home page which has a search bar that allows an administrator to browse the repository contents or even search for a particular object in the repository. During evaluation of the home screen, it was suggested that the main page have multiple records in a table with check boxes, which will allow administrators to remove or edit images. Figure 3.6 displays the page that allows an administrator to remove images from the repository. Version 2 allowed the administrator to specify an identifier for the object to be uploaded to the repository. However a better suggestion offered was to allow Fedora to handle the naming of objects, by generating a unique object identifier. Also, it was suggested that during the uploading of an object to the repository, the associated metadata be entered and stored along with the object.

3.4.4 Iteration 4 - Version 3(Refinement of final system)

Iteration 4 involved refining the system by incorporating all the user recommendations that were outlined in iteration 3. The system developed in iteration 3 was also evaluated by the staff at UCT Fine Arts Department. The evaluation chapter will discuss a case study evaluation of the system, where screenshots of the final system are shown.

The final implemented features of the implemented archive management system are:

- Login authentication of an archive administrator for the **BOLD** archive.
- Basic browsing of **BOLD** archive system contents.
- Single upload of an object, with associated metadata into the **BOLD** system archive.
- Deletion of an object from the **BOLD** archive system, using the unique object identifier as a key.

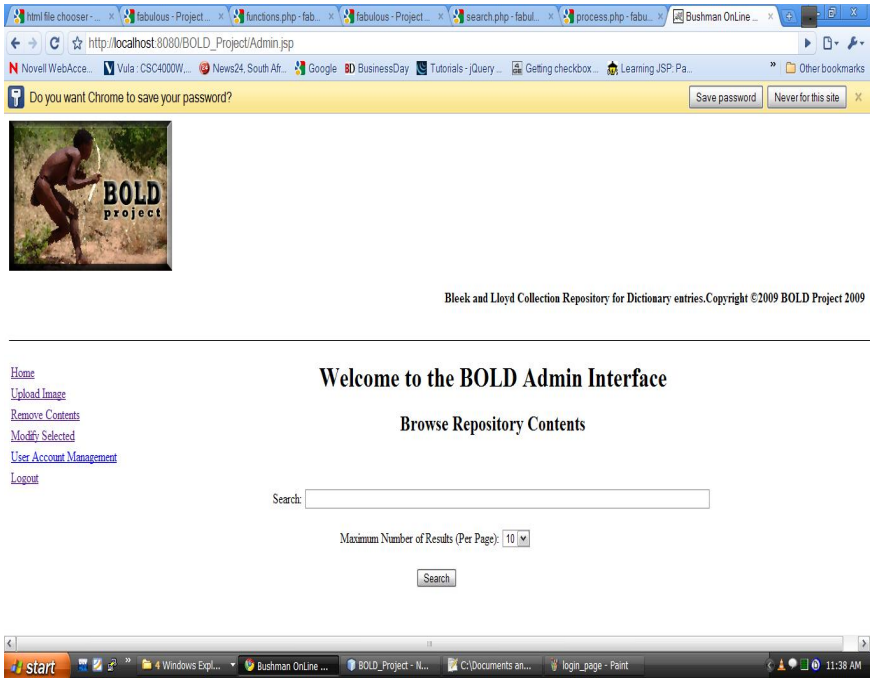

Figure 3.5: Version 2 - Administrator Interface Home page

Figure 3.6: Version 2 - Remove objects page (with checkboxes)

3.5 Implementation

Figure 3.7 displays a diagrammatic overview of the implementation of the archive management collection. The archive collection management was developed in 2 portions namely a back-end layer and a Web front-end layer, for the user interface. The system utilises the FedoraClient.jar interface provided by Fedora to implement various reposi-

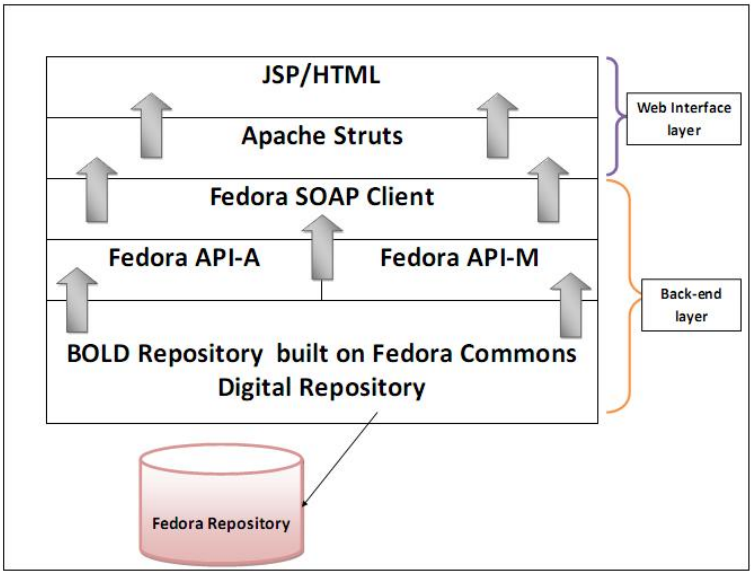

Figure 3.7: High level implementation view: Archive Management Collection

tory management functions. The Web user interface is implemented using Java Server Pages (JSP) to render the user interface.

3.5.1 Back-end layer implementation

A user-defined Simple Object Access Protocol (SOAP) client, using the Fedora API-A and API-M(enclosed in the FedoraClient.jar interface) was developed based on the requirements outlined in the requirements gathering phase of development,i.e.

1. Uploading of images(with associated metadata).
2. Deleting images from the repository given the unique PID(object identifier).
3. Editing information about the images (metadata editing).

The backend application uses the Fedora access and management API (API-A and API-M) to implement these user requirements.

3.5.2 Web-interface layer

The Web user-interface is rendered using Java Server Pages(JSP), built on top of the Apache Struts framework. Apache Struts is a free open-source framework for creating Java Web applications [Apache, 2009]. Apache Struts is a framework motivated by the notion of maintaining large dynamic web applications that can become very complicated when dealing with database code, page design code and control flow code on Java Server Pages(JSP).

The Struts framework solves this by adopting the Model-View-Controller (MVC) where:

- The *Model* represents the business logic. The model is responsible for providing the data from the database and saving the data into the data store.

- The *View* represents the user view of the application and is responsible for taking the input from the user, dispatching the request to the controller and then receiving a response from the controller and displaying the result to the user. HTML and JSPs are part of the view.
- The *Controller* represents the navigational code. The controller is responsible for receiving the request from the client. Once a request is received from client it executes the appropriate business logic from the Model and then produces the output to the user using the View component.

The Struts framework is designed to help developers create Web applications that utilise MVC architecture [Apache, 2009].

3.5.3 Development of API for other components

The search, browsing and display component of the **BOLD** archive makes use of the Fedora access API(API-A) to access the repository objects. The archive management system implemented an interface that manages collections of objects, that can be accessed by the searching and browsing module.

The **BOLD** image-based translation module does not interact with the Fedora repository that forms the basis of the **BOLD** archive. The translation module is connected to the archive management system using an API that defines a set of triggers to call methods(eg. data pre-processing, translation modules etc.) to initiate the functioning of the translation module.

3.6 Summary

In this chapter, the design and implementation of the archive administration system of the **BOLD** project, is discussed.
While every attempt was made to implement every single user requirement, some of the requirements were not implemented fully. These requirements were:

- Metadata editing.
- Batch upload of images via Java Server Pages(JSP).
- Management of multiple administrator accounts.
- Administrator User Account rights management.

Most of these functions were implemented at a low-level using the back-end layer but, due to time-constraints, the interaction with the user interface pages (Java Server Pages - Web interface) was not achieved. While not all of the required functionality was implemented, there was a reasonable amount of success in the implementation of the system.

Chapter 4

Evaluation

The development of the Bushman OnLine Dictionary followed an iterative model. As such testing was performed regularly during the various iterations of the project. Unit testing was a regular feature throughout the development of the system so as to mitigate potential bugs in the development of the system.

In this chapter an evaluation of the archive management system is discussed by referring back to the research hypothesis presented in the project proposal i.e. *Can we develop a useful and efficient archival system?* Evaluation of the final system was part of the fourth iteration of the development cycle.

4.1 Evaluation Plan

4.1.1 Scope

The primary purpose of evaluating the archive management system is to test whether the necessary requirements outlined in the project proposal and the requirements gathering discussed in the design chapter have been met.

4.1.2 Software Requirements

Evaluating the archive required a computer that had the Fedora Commons Digital repository version 3.2.1 installed on it with MySQL Server 5.1 as the underlying database. In addition, the following software was used in testing:

- Netbeans IDE 6.5.1(using Java JDK 1.5 or later).
- A Web browser (Mozilla Firefox and Google Chrome where used for the evaluation).
- A Web server. In this instance the default Apache Tomcat installation bundled with the Fedora installer was used for testing.

The performance tests where performed on a computer in the Computer Science Honours Laboratory with system specifications: Intel(R) Pentium(R) Dual-Core CPU E220@ 2.20 GHz 1.98 GB of RAM, running Windows XP Service Pack 3.

4.1.3 Test Requirements

3 tests were performed to evaluate the archive, namely:

- *Perfomance Testing*
  Determining how well the archive performs in managing batch operations such as ingest and deletion of objects from the repository.
- *Case Study Evaluation*
  The aim of the case study evaluation is to test the functionality of the archival system from the point of view of a repository administrator.
- *Usability Testing*
  How well do users (experts in information management) respond to the system? This relates to how users interact with and use the archive management system.

4.2 Performance Testing

Performance testing is of paramount importance in an application that has to handle a large amount of data. One of the primary features of the Fedora digital repository is the ability to scale to a million objects [Fedora, 2009].

The performance testing analyses the behaviour of the batch ingest and delete operations implemented in managing the archive, with the primary purpose being to test the research hypothesis of the efficiency of the **BOLD** archive. The batch operations tested in the performance testing, were developed by the developer and implemented to manage the back-end layer, using the Fedora repository access and management API(API-A and API-M). Figure 4.1 gives an overview of where the performance testing is performed in the archive management system.

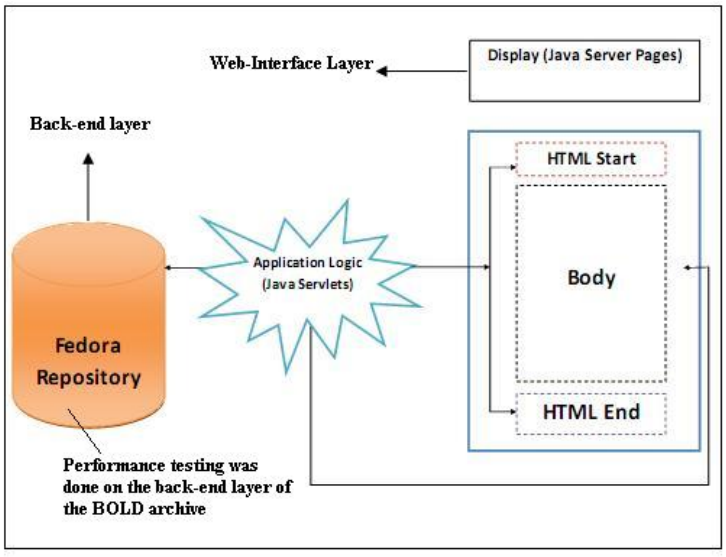

Figure 4.1: Portion of archive management system that is tested

4.2.1 Experiment 1 - Batch Ingest of object into the BOLD archive collection system

An archive administrator of the **BOLD** project may want to upload multiple images to facilitate more efficient management of the archive. For the back-end layer a batch object ingest functionality was created, with the specific purpose of uploading the TIFF images corresponding to the dictionary images.

- *Aim*
  To test the performance of the batch ingest implementation of the archive management system, and to analyse its progression as the size of the batches increases.

- *Scale of Experiments*
  The original TIFF format images, located on the local server (mufasa.cs.uct.ac.za) situated in the Computer Science department, were used as a test subset for the performance test. The reason for this choice was twofold - firstly, the images were readily available on the department server and secondly the original images were chosen to give an accurate representation of the performance of the batch upload implementation.

  Test subsets of n-images were used to test the batch ingest feature using n = [10, 50, 100, 250,500 and 1000].

- *Method*

  – A separate Java application(created solely for testing purposes and connected to the archive management system) was created for performance testing, using the SOAP client developed for the back-end layer of the system.
  – Time measurement of the batch ingest was recorded using the Java measurement of time - System.currentTimeMillis ().
  – A subset of test images was created, by randomly choosing subsets from the server.
  – The application was executed and the time taken to ingest the subset of n - images was calculated and recorded in a file. The test was executed for n = [50, 100, 250, 500 and 1000].
  – The images were ingested from a specified directory, calling the buildObject method that ingests and commits an object to the repository.

- *Results*
  The results of the batch ingest are tabulated below, with a graph summarising the performance of the batch ingest implementation shown in Figure 4.2:

- *Analysis of Results*
  The results clearly show constant increase in ingest time as the number of objects ingested increases.

  From an archive administrator perspective, the average ingest time per object using the batch ingest implementation is approximately 0.259 seconds per image(259.857ms). The average ingest time per object is tabulated for all test subsets in Table 4.1.

Batch Ingest Size(n)	Ingest Time(ms)	Average Ingest Time Per Object(ms)
10	2781	278.1
50	13235	264.7
100	25844	258.44
250	57750	231
500	122156	244.312
1000	241594	241.594

Table 4.1: Results of batch ingest performance test

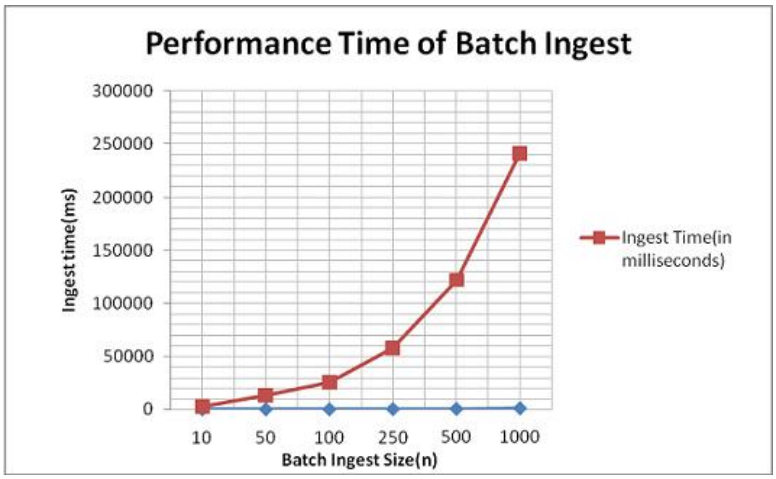

Figure 4.2: Graph of batch ingest performance

In addressing the research question of an efficient archive, moderate success was achieved in the implementation of batch ingest. This is in comparison to other testing frameworks developed by Fiz Karlsruhe [Karlshure, 2009] that develops test frameworks for the Fedora Commons digital repository. The Fiz Karlsruhe testing framework for the batch ingest process reported an ingest rate of 10 objects per second [Karlshure, 2009]. The batch ingest operation implemented for the **BOLD** archive management system ingested an object every 0.25954 seconds, which works out to 2.594 objects per second.

Ingest time for an object added to the repository depends on the binary content of the object. In this instance, the set of dictionary scans are in TIFF format which has a large binary encoding relative a JPEG-encoded image.

- *Conclusion*
  The implementation of batch ingest for the Bushman OnLine Dictionary worked reasonably well for this particular project. We notice a constant increase in the ingest time as the batch size increases, indicating linear scalability.

4.2.2 Experiment 2 - Batch Deletion of objects from the BOLD collection system

An archive administrator of the BOLD project may want to delete multiple images at the same time to facilitate more efficient management of the archive. For the back-end

layer a batch object delete function was implemented using the Fedora Management API(API-M) method purgeObject to remove objects from the BOLD archive.

- *Aim*
  To test the performance of the batch delete implementation of the archive management system and to analyse how long it takes to delete images from the repository as the number of images increases.

- *Method*
  - As per the previous experiments, the test images were ingested into the BOLD repository for subsets of n - images where n = [10,50,100,250 and 1000].
  - After ingest, a batch delete operation was performed. In this operation, all of the images were deleted from the repository using the purgeObject method implemented in the back-end layer. (For this performance test, batch deletions of multiple images were considered only).

- *Results*
  Results of the batch delete operation are shown in Table 4.2. Figures 4.3 and 4.4 graph the batch delete performance for all the test subsets, and the average delete time.

Batch Ingest Size(n)	Delete Time(ms)	Average Delete Time(ms) Per Object
10	1156	115.6
50	5360	107.2
100	10344	103.44
250	24828	99.312
500	48796	97.592
1000	97250	97.25

Table 4.2: Results of batch delete performance test

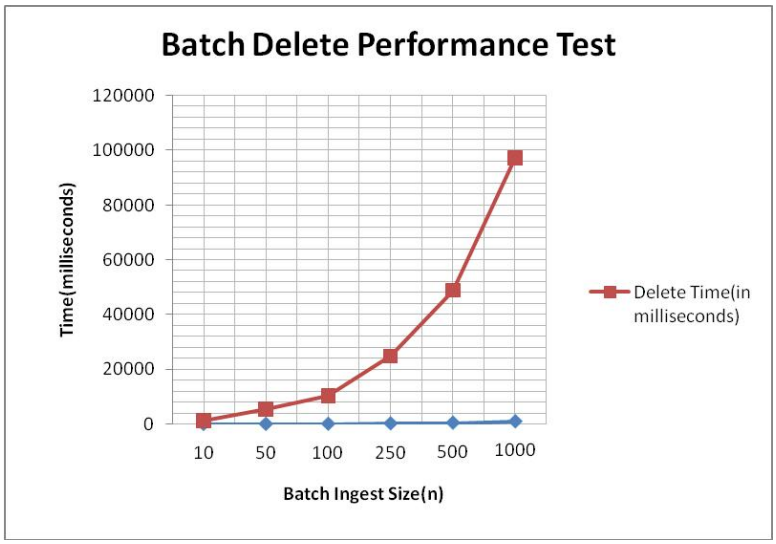

Figure 4.3: Batch delete performance graph

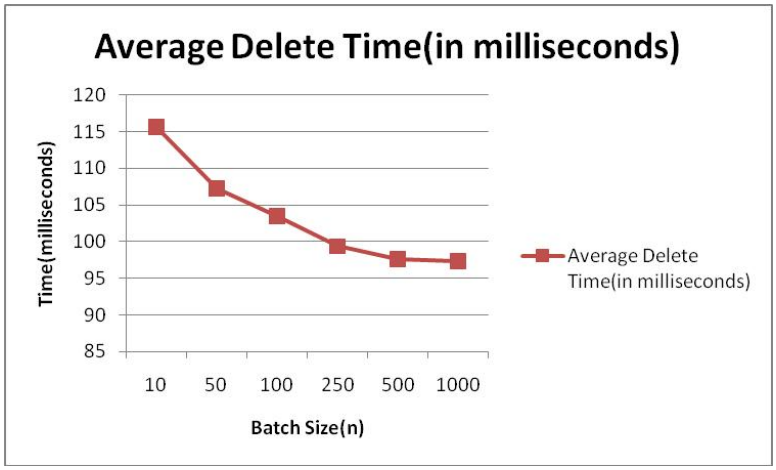

Figure 4.4: Average delete time

- *Analysis of Results*
  Batch delete times increase at a constant rate as the batch input size n increases, thus indicating that there is no deterioration in performance as the number of objects deleted from the repository increases.

  It would appear that as the number of objects purged from the archive increases, the performance of the batch delete improves accordingly - as shown in Table 4.2. Table 4.2 also shows that on average it takes 0.1064 seconds to delete an object from the repository.

- *Conclusion*
  This experiment looked at the batch delete operation implemented for the archive and showed it to perform well in the purging of objects from the repository.

4.3 Case Study Evaluation

The case study evaluation is used to perform a test of the system with a real set of data - in this case the original scanned TIFF images obtained from the Center for Curating the Archive. The aim is to test the functionality of the archival system from the point of view of the repository administrator.

- *Aim*
  To perform several tasks that are typically associated with managing the Bushman OnLine Dictionary repository. The aim is to test the functionality of the repository using real-world data, in this instance the original TIFF images.

- *Method*
  Use Case 1: Administrator Login
  Actor: Archive Manager
  Description: By clicking a link from the homepage of the Web application, an administrator should be taken to an administrative portal allowing them to login to the administrator page.

  Use Case 2: Adding an object to the archive system

Actor: Archive Manager
Description: The administrator should be able to add an object to the archive. On clicking the link to submit, the user will upload a file to ingest, add the associated Dublin Core metadata and submit the request to the backend application through the Java Server Pages(JSP) for processing.

Use Case 3: Deleting an object from the archive
Actor: Archive Manager
Description: An administrator must be able to check one or more images and submit the request via a Java Server Page that communicates with the repository. The object(s) will then be removed from the repository.

Use Case 4: Logout of the system
Actor: Archive manager
Description: An administrator must be able to log out of the archive at any time.

- *Results*
  Screenshot results of the use case evaluations are shown in Figure 4.5, Figure 4.6, Figure 4.7 and Figure 4.8:

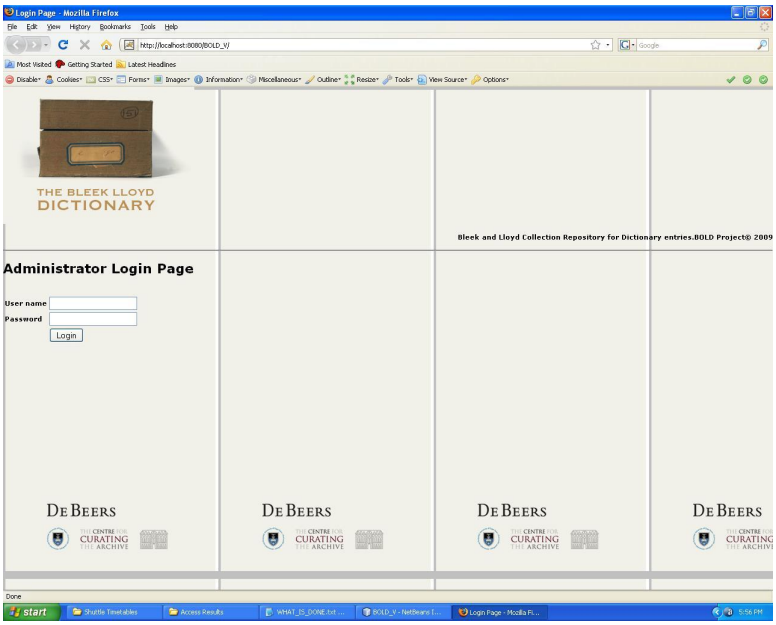

Figure 4.5: Use Case one screenshot - Login

It must be noted that during the ingest of some of the TIFF images, unexpected bugs occurred probably due to the byte encoding that the Java Server Pages(JSP) file uploader uses, to enable the upload of images to the BOLD repository.

When a file is submitted for upload to the repository via the JSP, the client browse locates the file and sends it using HTTP POST(the file is encoded) to the servlet that is responsible for processing the file upload(in this case, the servlet implemented handled the upload to the **BOLD** archive. Once the file reaches the servlet it processes the HTTP POST data to extract the encoded file for processing. This is where the error appeared, and it was discovered that there is no method in the

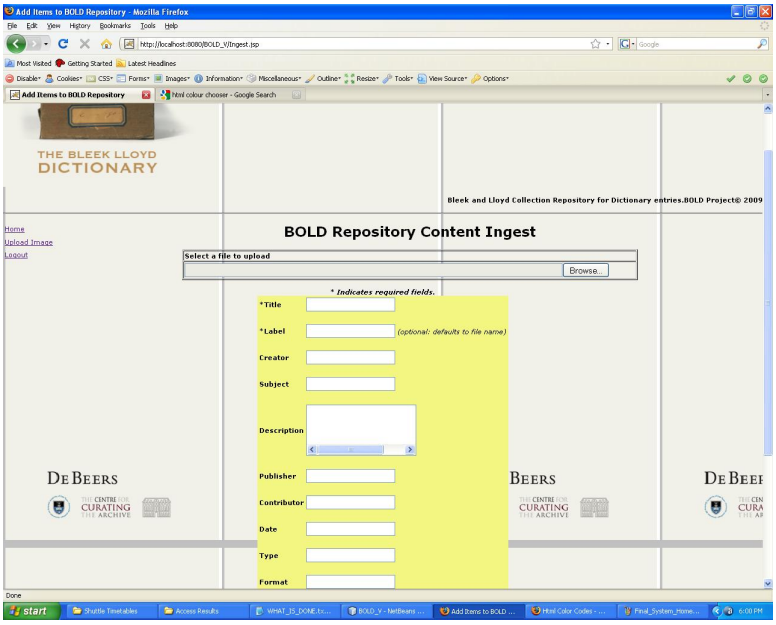

Figure 4.6: Use Case two screenshot - Add images

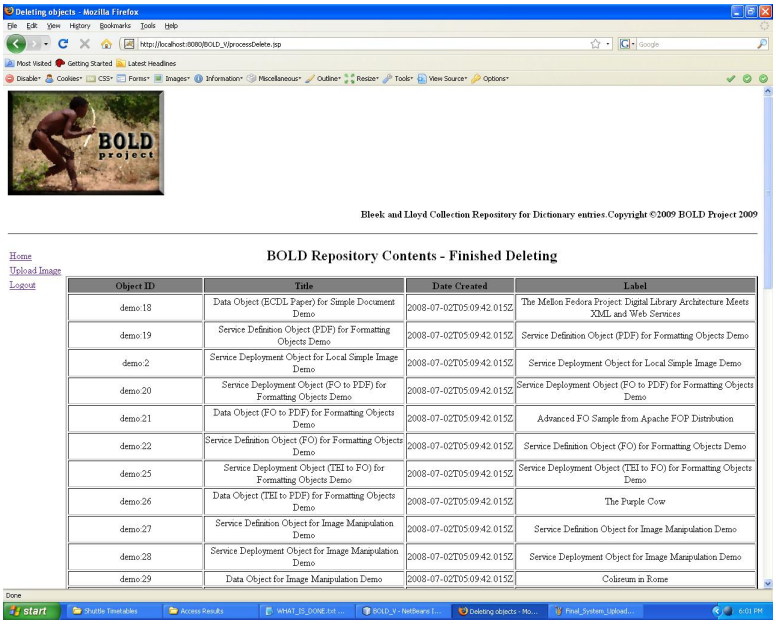

Figure 4.7: Use Case three screenshot - Delete images

Java servlet API that can extract the content of the encoded file - this is the developer's responsibility to implement. While the implementation worked on most of the TIFF images, it was not discovered why the file extraction failed on some of the images.

- *Conclusion*
  The case study evaluation was performed without any major errors or hiccups. Not at all 16 000 images in the Bleek and Lloyd Collection were not ingested, due to time constraints. Instead, a subset of 50 of the original TIFF images where in-

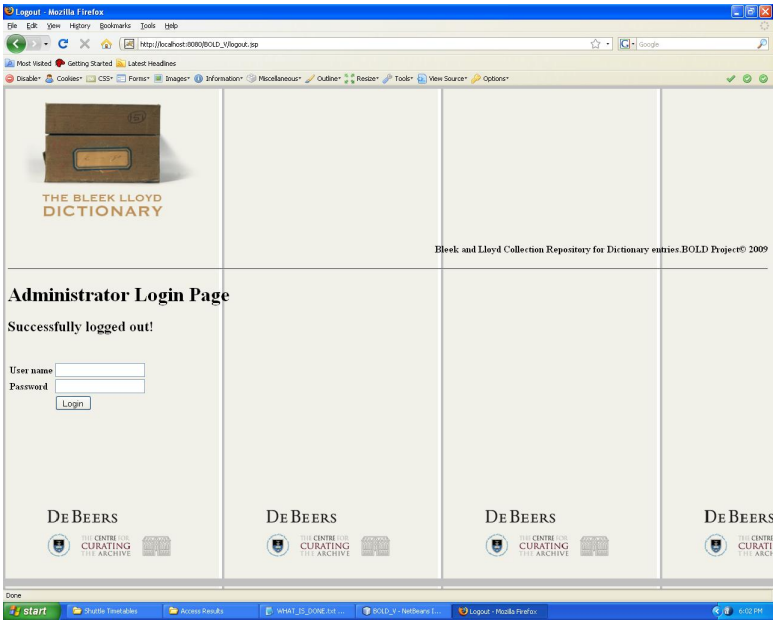

Figure 4.8: Use Case four screenshot - Logout

gested into the repository.

4.4 Usability Testing

- *Aim*
  To test the usefulness of the archival system implemented, a usability test was performed. Usability tests were performed with a variety of users - with a strong bias towards users with some sort of expertise or knowledge in managing large collections of information.

- *Usability Standards*
  ISO 9241-11 suggests that measures of usability should cover [Brooke, 2009]:

  – Effectiveness (the ability of users to complete tasks using the system, and the quality of the output of those tasks),

  – Efficiency (the level of resource consumed in performing tasks),and

  – Satisfaction (users' subjective reactions to using the system).

- *Users*
  The initial plan for usability testing was to test the system on users who have some experience in archive management or managing large collections of information. Example of such users include:

  – UCT Fine Arts department lecturers (target end users of the system) .

  – Students from Digital Libraries Laboratory (expert users).

  – Selected computer science students (high computer literacy students).

  - Users who manage large collections of information e.g. Photo collections, documents, electronic journals.

  Due to time constraints, only 10 people were used for usability testing. The spread of the users was:

  - 1 non-UCT affiliated user (a librarian at Mowbray public library).
  - 4 Computer Science students in 2nd and 3rd year (high computer literacy).
  - 5 students from across the various faculties.

- *Method*
  User evaluation involved a hands-on interaction with the archive management component of the project. The evaluation involved a 30 minute task-based one-on-one session where users had the opportunity to interact with the archive management system. The users had 15 minutes to perform the given task and at the end of the 15 minutes time frame or completion of the tasks, they filled in a questionnaire relating to the usability of the system. The usability questionnaire used for this experiment is an adaptation of an existing usability questionnaire Davis [1989].

- *Results and Analysis of Results*
  Results from the usability questionnaire are included in the appendix to this report. Figure 4.9 graphs the usability results data. Questions that were asked in the usability questionnaire:

  1. I was able to upload the picture and add its associated metadata, to the repository.
  2. I was able to view the image once I had added the picture to the repository.
  3. I was able to delete the image successfully from the repository.
  4. I was able to login and logout from the system successfully.
  5. It was easy to navigate the system.
  6. I was able to recover from errors made quickly and easily.
  7. The interface has enough information to facilitate user navigation with confidence.
  8. My interaction with the system was clear and understandable.
  9. Using the system would make it easier to manage large collections of information.
  10. Using the system would improve my performance in managing large collections of information.

  (One of the questionnaires filled in could not be considered due to a power failure that occurred during testing). The test questionnaire had 2 sections - the first section(questions 1 - 5) focused on the task flow when utilising the archive management system, while the second portion of the questionnaire(questions 6 - 10) focused on the perceived ease of using the archive management system.Each question was phrased in such a way that agreements correspond to positive feedback about the archive management system, with disagreements being indicative of negative feedback. From the distribution of the results, there were mixed responses about the usability of the system. Testing revealed a few bugs not picked

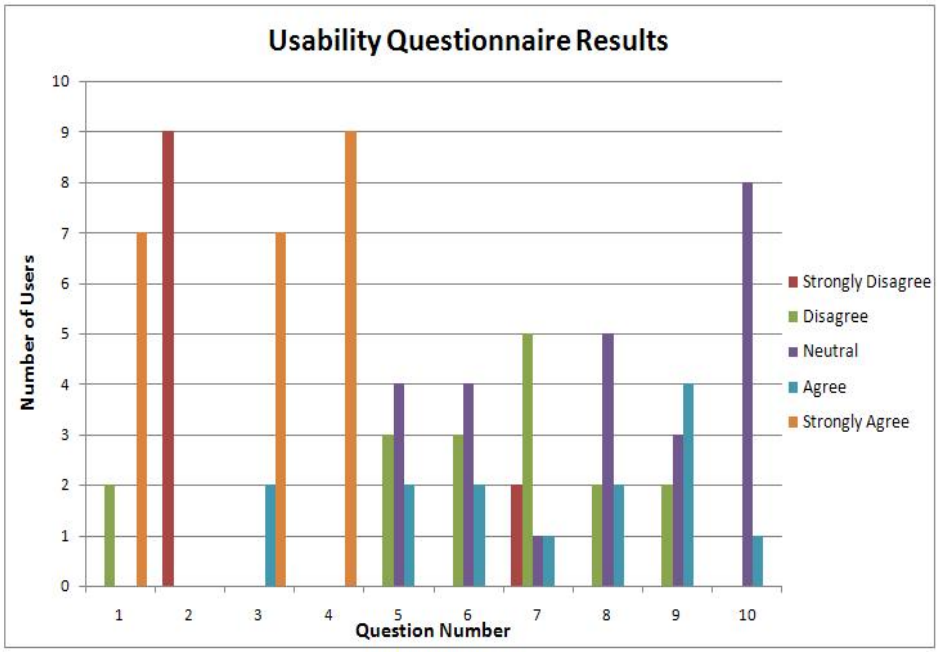

Figure 4.9: Results from usability testing

up during unit testing such as slow system response when deleting images as well as insufficient error handling when users entered metadata about the image. Users also complained about the administrator interface as being too dull and uninteresting.

From a task-centric point-of-view, users where able to perform the tasks with general ease - with the exception of question 2. Upon analysis, it was later found that the question was a bit vague as users did not know that they had to return to the home screen to see the image that they had uploaded.

It must be noted that the results obtained from the usability test are only indicative of the general system usability. To adequately test the system and measure usability, a more thorough evaluation would need to be performed with the end users at the Center for Curating the Archive and archive managers .

- *Conclusion*
  The usability testing, though limited in scope, revealed that the implementation of the **BOLD** archive system had a few system flaws. Although usable, there is scope for improvement based on comments from the usability questionnaire results.

4.5 Summary

Three tests where performed to answer the research question of building an efficient and useful archive management system. The performance test revealed the implementation of the **BOLD** archive to be efficient with respect to the batch ingest and deletion of images from the archive. The batch ingest implementation revealed that there was linear scalability, based on the test performed on the test subset, while the batch delete

operation performed even better as the number of objects deleted increases.

The usability testing met the research hypothesis of building a useful archive management system, but also revealed some flaws which where not accounted for during unit testing.

Chapter 5

Conclusion

The Bushman OnLine Dictionary(BOLD) project is an image-based dictionary with the key aim of integrating the recent set of digital scans into a dictionary with the existing Bleek and Lloyd Collection. The system has 3 sub-components namely: an archive management system; a search, browsing and display functionality; as well as an image-based translation module.

The archive management system was implemented using the open-source Fedora Commons digital repository software for the back-end layer, with the Web interface design using Java Server Pages(JSP). Numerous challenges were encountered in using the Fedora API, due to the lack of a developer's guide and ambiguous documentation but these challenges were solved relatively well, with the necessary guidance and expert advice.

The aim of the archive system was to meet the research question of building an efficient and useful archive. There was moderate success in addressing this research question, as revealed by the evaluation of the system. The archive system performed fairly well in addressing the efficiency part of the research question, but evidence from the usability testing showed that more work was needed to make the archive system more usable - even though most of the functionality was implemented.

While the main research aim of this project of building a *reusable, generic archival system that allows users to access an image-based dictionary* was not accomplished, the archive management system managed to succeed in addressing the usable and efficient archive research aim. There is scope for more work in adding additional features to the Bushman OnLine Dictionary (BOLD) archive management system, and it is believed that this project has laid a solid platform for future extensions.

Chapter 6

Future Works

6.1 Batch utility services

As at present, the **BOLD** archive allows for single object manipulation, meaning that only one object can be ingested at a time from the Web interface, one object can have its metadata edited. Batch utility services such as: i.e

- Batch upload of images,
- Batch metadata editing, and
- Batch deletion of images.

are a possible extension to this project. Numerous projects exists that have already attempted to address this issue including the *FABULOUS* - Fedora Arrow Batch Utility with Lots of User Services project[Forum, 2009] initiated by the University of South Australia, as well as the *Elated* architecture(A management and search front-end to the Fedora project) [Elated, 2009].

6.2 User Interface

At present, the system makes use of Java Server Pages(JSP) technology to render the user interface. The archive management system could incorporate the use of AJAX technologies to facilitate a more enriching and dynamic user interface.

6.3 Thumbnail previews

A useful feature that the **BOLD** archive management system could have, is a thumbnail view of the images stored in the archive. This could also be extended to pop-up windows created with Javascript to view the scanned dictionary images possibly with the descriptions and metadata of the image.

6.4 User Account Management

Currently the **BOLD** archive management system uses only one administrator account(the default administrator account created during the installation of the Fedora repository), to manage the system. A future extension to this project could allow multiple administrators with associated rights, to manage the archive.

6.5 Searching and Browsing functionality

Research in existing archive management systems, shows that most of them have some form of searching and browsing functionality. This feature can be very useful to an archive manager, that manages a repository with a substantial amount of objects.

Bibliography

Aluka. Aluka - lloyd and bleek collection. http://www.aluka.org/action/showCompilationPage?doi=10.5555/AL.CH.COMPILATION.COLLECTION-MAJOR.LBC&cookieSet=1, October 2009. Last updated - 2009.

Apache Software Foundation Apache. Apache struts - welcome. http://struts.apache.org/, September 2009. Last Updated - 09/30/2009.

Titus Barik. Introducing the java content repository api. http://www.ibm.com/developerworks/java/library/j-jcr/, October 2009. Last updated - 27/06/2006.

P. Baudoin and M. Branschofsky. Implementing an institutional repository: The dspace experience at mit. *Science and Technology Libraries*, 24(1/2):31–45, 2003.

F. J. Brooke. Sus - a quick and dirty usability scale. http://www.usabilitynet.org/trump/documents/Suschapt.doc, October 2009. Last updated - N/A.

CCA. Centre for curating the archive at the university of cape town(uct). http://cca.uct.ac.za//, October 2009. Last updated - March 2009.

F. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology., 1989. Last updated - N/A.

Reim Doumat, Elöd Egyed-Zsigmond, Jean-Marie Pinon, and Emese Csiszar. Online ancient documents: Armarius. In *DocEng '08: Proceeding of the eighth ACM symposium on Document engineering*, pages 127–130, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-081-4. doi: http://doi.acm.org/10.1145/1410140.1410167.

DSpace.org. Dspace - open source software. http://www.dspace.org/, October 2009. Last updated - 2/11/2009.

North Carolina ECHO. North carolina echo, exploring cultural heritage online. http://www.ncecho.org/about/index.shtml/, May 2009. Last updated - 31/10/2009.

Elated. Elated: a general-purpose web-based client for the fedora repository system. http://elated.sourceforge.net/, October 2009. Last updated - 2006.

Fedora Commons Fedora. Fedora digital object model. http://www.fedora-commons.org/documentation/3.0b1/userdocs/digitalobjects/objectModel.html, October 2009. Last updated - 3/11/2009.

Fedora Commons Developer's Forum. Fedora tools - fedora commons developer's forum - duraspace wiki. http://www.fedora-commons.org/confluence/display/DEV/Fedora+Tools, October 2009. Last updated - 2008.

Greenstone. Greenstone digital library software. http://www.greenstone.org/, May 2009a. Last updated - N/A.

Wikipedia Greenstone. Greenstone (software) - wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Greenstone_(software), October 2009b. Last updated - 10/10/2009.

Open Archives Initiative. Open archives initiative protocol for metadata harvesting. http://www.openarchives.org/pmh/, May 2009. Last updated - N/A.

Fiz Karlshure. Fiz karlsruhe: Advancing science (docs: main). http://fedora.fiz-karlsruhe.de/docs/, October 2009. Last updated - 21/05/2008.

Brian. Lavoie and L. Dempsey. Thirteen ways of looking at...digital preservation. *D-Lib Magazine*, 10(7/8), 2004. ISSN 1082-9873.

Jyi-shane Liu and Mu-Hsi Tseng. Mediating team work for digital heritage archiving. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 259–268, New York, NY, USA, 2004. ACM. ISBN 1-58113-832-6. doi: http://doi.acm.org/10.1145/996350.996414.

Bertram Ludäscher, Richard Marciano, and Reagan Moore. Preservation of digital data with self-validating, self-instantiating knowledge-based archives. *SIGMOD Rec.*, 30 (3):54–63, 2001. ISSN 0163-5808. doi: http://doi.acm.org/10.1145/603867.603876.

Sanvir Manilal. The bold project - searching and browsing the bushman on-line dictionary. Technical report, Department of Computer Science, University of Cape Town, 2009.

Gary Marsden, Katherine Malan, and Edwin Blake. Using digital technology to access and store african art. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 528–529, New York, NY, USA, 2002. ACM. ISBN 1-58113-454-1. doi: http://doi.acm.org/10.1145/506443.506464.

Reagan W. Moore and Richard Marciano. Building preservation environments. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 424–424, New York, NY, USA, 2005. ACM. ISBN 1-58113-876-8. doi: http://doi.acm.org/10.1145/1065385.1065529.

Shalaka Natu and John Mendonca. Digital asset management using a native xml database implementation. In *CITC4 '03: Proceedings of the 4th conference on Information technology curriculum*, pages 237–241, New York, NY, USA, 2003. ACM. ISBN 1-58113-770-2. doi: http://doi.acm.org/10.1145/947121.947175.

Theo Nicolakis, Carlos E. Pizano, Bianca Prumo, and Mitchell Webb. Protecting digital archives at the greek orthodox archdiocese of america. In *DRM '03: Proceedings of the 3rd ACM workshop on Digital rights management*, pages 13–26, New York, NY, USA, 2003. ACM. ISBN 1-58113-786-9. doi: http://doi.acm.org/10.1145/947380.947384.

Wikipedia OAIS. Open archival information system. http://en.wikipedia.org/wiki/Open_Archival_Information_System, May 2009. Last updated - 28/10/2008.

OCLC. Oclc global gateway [oclc]. http://www.oclc.org/uk/en/global/default.htm, May 2009. Last updated - N/A.

Lagoze C. Payette S. Flexible and extensible digital object and repository architecture (fedora). In *Research and Advanced Technology for Digital Libraries, Second European Conference, ECDL '98*, pages 45–59, 1998.

Joyce Ray, Robin Dale, Reagan Moore, Vicky Reich, William Underwood, and Alexa T. McCray. Panel on digital preservation. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 365–367, New York, NY, USA, 2002. ACM. ISBN 1-58113-513-0. doi: http://doi.acm.org/10.1145/544220.544313.

Pippa Skotnes. The digital bleek and lloyd. `http://lloydbleekcollection.cs.uct.ac.za/`, May 2009. Last updated - N/A.

Lisa Spiro, Marie Wise, Geneva Henry, Chuck Bearden, Sid Byrd, Eva Garza, and Michael Decker. Enabling exploration: travelers in the middle east archive. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 163–164, New York, NY, USA, 2006. ACM. ISBN 1-59593-354-9. doi: http://doi.acm.org/10.1145/1141753.1141784.

Hussein Suleman. *Digital Libraries Without Databases: The Bleek and Lloyd Collection*, volume 4675/2007. Springer Berlin / Heidelberg, 2007. doi: 10.1007/978-3-540-74851-9_33.

Kyle Williams. The bold project - the bold translator. Technical report, Department of Computer Science, University of Cape Town, 2009.

D. Zorich. *A Survey of Digital Cultural Heritage Initiatives and Their Sustainability Concerns*. Council On Library and Information Resources, Washington D.C, 2003.

Appendix A

Usability Questionnaire and Results

University of Cape Town

Computer Science Department

CONSENT AND WAIVER

TITLE: Evaluating the usefulness of the Bleek and Lloyd dictionary archive management component.

PURPOSE: The purpose of this research is to determine how useful and usable the archive that manages the Bleek and Lloyd dictionary collection is.

REQUIREMENTS FOR EVALUATOR: You will be given a task list that involves performing certain functions associated with archive management. Thereafter, you will be required to fill in a questionnaire.

TIME REQUIRED: Maximum of 30 minutes

PRIVACY: Your identity will be kept confidential. Your name will not be used in any report. All results and opinions given here are purely for research purposes.

VOLUNTARY PARTICIPATION: Your participation in this study is voluntary. You have the right to withdraw from this study at any time without consequence.

If you have any questions or queries, then please feel free to contact me:

Lebogang Molwantoa
lmolwantaa@cs.uct.ac.za

_____

TO WHOM IT MAY CONCERN:

I hereby grant full permission to take notes of my comments during the usability test for Bushman

OnLine Dictionary (BOLD) Project.

Signature: _____

Date: _____

PRE-TEST QUESTIONNAIRE

Demographic Information Questionnaire

(Please note, your information will not be sold or given to outside entities. It is for internal use only.)

Age: _____

Gender: Male/Female

1. How often do you use the internet? (Circle the most appropriate)

Daily Weekly Monthly Occasionally Never

2. Do you have any experience in managing an archive/large collections of data such as a photo collection?

Yes/No

3. How do you usually manage large collections of information such as photos, articles, documents?

_____

_____

4. Have you heard about the Bleek and Lloyd Collection? YES/NO

BOLD Project user evaluation questionnaire - Task List

Instructions

The Bleek and Lloyd Collection is a set of documents that preserve the history and culture of the early
Bushman inhabitants of the Western Cape of South Africa. This project aims to integrate a collection of
digital scans corresponding to a dictionary to the existing Bleek and Lloyd Collection. This dictionary may
be used to interpret and understand the original text. The project also aims at providing a framework for
future image-based dictionaries, aimed at cultural preservation.

On the web page opened, login using the following:
User name: fedoraAdmin
Password: cancer

Once logged in, feel free to navigate the administrative archive and explore the various links and
functions provided on the home page.

Next perform the following tasks:

1. Navigate to the upload image link and upload a file located on the desktop called My_Pic.jpg.
   Add any metadata that you deem appropriate to describe the image. Click on the submit button for the object to be uploaded.
2. Navigate back to the home page and browse to the image that you have just uploaded. Save the changes.
3. Navigate back to the home page to the same image. Select the checkbox and delete the image.
4. Logout.

Feel free to do a more thorough evaluation of the system and explore more functionality provided on the admin interface.
When you are done with the task set please complete the questionnaire below.
If you have any questions, please do not hesitate to ask me.

Thank you.

Perceived Usefulness and ease of use questionnaire

SECTION A – General Task-specific Questions

1)
I was able to upload the picture and add its associated metadata, to the repository.

1	2	3	4	5
Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

2) I was able to view the image once I had added the picture to the repository.

1	2	3	4	5
Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

3)
I was able to delete the image successfully from the repository.

1	2	3	4	5
Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

4) I was able to login and logout from the system successfully.

1	2	3	4	5
Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

5) It was easy to navigate the system.

1	2	3	4	5
Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

6) I was able to recover from errors made quickly and easily

1	2	3	4	5
Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

SECTION B - Perceived usefulness & perceived ease of use

1. My interaction with the system was clear and understandable.

1	2	3	4	5
Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

2. Using the system would make it easier to manage large collections of information.

1	2	3	4	5
Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

3. Using the system would improve my performance in managing large collections of information.

1	2	3	4	5
Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

4. I would find the system useful in the task of managing large collections of information such as photos and documents.

1	2	3	4	5
Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

5. Learning to operate an archive for managing large collection of information would be easy for me using the system.

1	2	3	4	5
Strongly disagree	Disagree	Neutral	Agree	Strongly Agree

8. Did you encounter anything unexpected during the evaluation (bugs, lags in performance etc)?

_____

_____

_____

9. General Comments about the archive system?

_____

_____

_____

Results from Usability Questionnaire

QUESTION	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1. I was able to upload the picture and add its associated metadata, to the repository.		2			7
2. I was able to view the image once I had added the picture to the repository.	9				
3. I was able to delete the image successfully from the repository.				2	7
4. I was able to login and logout from the system successfully.					9
5. It was easy to navigate the system.		3	4	2	
6. I was able to recover from errors made quickly and easily		3	4	2	
7. The interface has enough information to facilitate user navigation with confidence.	2	5	1	1	

QUESTION	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
8. My interaction with the system was clear and understandable.		2	5	2	
9. Using the system would make it easier to manage large collections of information.		2	3	4	
10. Using the system would improve my performance in managing large collections of information.		8	1		
Learning to operate an archive for managing large collection of information would be easy for me using the system.		3	6		