A Lightweight Digital Library Framework

Lighton Phiri
<lphiri@cs.uct.ac.za>

Supervised By
Hussein Suleman
<hussein@cs.uct.ac.za>

Master of Science Research Proposal

Department of Computer Science
Science Faculty
University of Cape Town

October, 2011

1 Introduction

Digital Libraries are information systems that store digital objects, and have associated services for accessing, managing, and preserving the digital objects.

Digital Libraries began as an abstraction layered over databases to provide higher level services. As the services and tools became more complex, they became more difficult to maintain, extend and reuse. This effectively raises the preservation costs and thus poses a challenge in environments with limited resources. Organisations that are faced with such resource limitations include cultural heritage organisations and a slew of other organisations in developing countries in regions such as Africa.

A number of currently existing solutions potentially rely on the availability of adequate Internet bandwidth for them to function adequately. This poses a challenge for digital collections that are hosted in regions where Internet bandwidth is both scarce and expensive.

One possible solution is to explicitly avoid formalisms, abstractions and Application Programming Interfaces (APIs) so that Digital Library Systems are more easily adopted and managed over time. This research thus proposes the design of a lightweight Digital Library architecture that may result in Digital Library Systems that may be easily adopted and managed.

2 Related Work

Research in the field of Digital Libraries has resulted in a number of different frameworks and architectural designs for Digital Library Systems. The variation in the designs can be largely attributed to the different design goals and specific problems that the systems were initially designed to address.

2.1 Frameworks

2.1.1 DELOS DL Reference Model

The DELOS Digital Library reference model was initiated on the premise that the Digital Library universe was a complex domain that could not be captured using a single definition.

The reference model identifies three different systems operating within the Digital Library universe: a Digital Library (DL) representing an organisation that collects manages and preserves digital content; a Digital Library System (DLS) for implementing DL facilities; and a Digital Library Management System (DLMS) comprising of tools for administering the DLS.

2.1.2 Kahn/Wilensky Framework

This is a generic information system framework for distributed digital object services with digital objects as the main building blocks. The framework is based on an open architecture that supports large and distributed digital information services [6].

The Kahn/Wilensky framework has been used as a model for implementation of Digital Library Systems [8], however, it describes an infrastructure that is too general-purpose and does not provide enough detail on how to implement specific digital library concepts such as long term preservation.

2.1.3 5S Framework: Streams, Structures, Spaces, Scenarios and Societies

The 5S framework is based on formal definitions and abstraction of five fundamental concepts [4]:

- Streams: sequences of elements of arbitrary type such as bits, characters or images, that can be used to model static or dynamic content.
- Structures: well-structured information objects, taxonomies, system connections or relationships to facilitate organisation of parts of a whole.
- Spaces: sets of objects, with their associated operations.
- Scenarios: sequence of events that may have parameters.
- Societies: sets of entities and relationships

2.2 Digital Library Software Tools

There are a number of proprietary and free and open source (FOSS) digital library software tools that have been developed. FOSS tools are the most commonly used and a discussion of three popular ones follows.

2.2.1 CDSware/CDS Invenio

CDS Invenio, formally known as CDSware, is an open source repository, developed at CERN[1] and originally designed to run the CERN document server[2]. CDS Invenio provides an application framework with necessary tools for building and managing a Digital Library System [10, 14].

CDS Invenio was designed to run on GNU/Unix systems, with a MySQL database backend server and Apache/Python Web application server. The architecture is made up of modules, with specifically defined functionalities, that interact with one another, the database and the interface layers .

2.2.2 DSpace

DSpace is an open-source repository software that was specifically designed to store digital research and institutional material produced by an organisation or institution [13]. The architectural design was largely influenced by the need for material to be stored and accessed over a long time.

DSpace is organised into a three-tier architecture, composed of: an application layer; a business logic layer; and a storage layer. The storage layer stores digital content within an asset store --a designated area within the operating system's file system; or can alternatively use a storage resource broker (SRB). The digital content file locations and corresponding metadata are stored within a relational database management system (RDBMS) [13]. It can be argued that this architectural design can make it difficult to re-create the objects in the event

[1] http://www.cern.ch/

[2] http://cdsweb.cern.ch/

of a problem, since one would have to match the digital contents with the corresponding metadata in the RDBMS.

Furthermore, DSpace's monolithic architecture makes it difficult for it to be adopted --a detailed usability study by Körber and Suleman [7] identified administrative usability problems associated with DSpace.

2.2.3 EPrints

EPrints[3] is a generic archive software tool designed to create highly configurable Web-based archives. It was developed to help foster open access to research publications and provides a flexible Digital Library platform for building repositories [5].

EPrints runs within an Apache HTTP server and uses a MySQL database server to store metadata about records and users. The actual files in the archive--e-prints--are stored on the file system.

2.2.4 Fedora

Fedora is an open-source Digital content repository service designed for managing and delivering complex digital objects [3]. Fedora was designed to handle any type of digital content and its key strength is its support for preservation and standards.

The Fedora architecture is based on the Kahn and Wilensky framework [4] with a distributed model that makes it possible for complex digital objects to make reference to content stored on remote storage systems. Unlike DSpace, however, digital objects in Fedora are stored in a data store, on the file system. Rebuilding the repository is easier, since both the digital content and metadata are stored on the file system. A metadata registry, in form of an RDBMS, is required to store metadata required for searching.

However, Fedora is complex and lacks a dedicated user interface - a characteristic that has led to further development of Web interface tools such as Fez[4] and Islandora[5].

2.2.5 Greenstone

Greenstone is an open-source software tool that was designed for building and distribution of digital collections. The software's ability to redistribute collections on a self-installing CD-ROM has made it a popular tool in regions with very slow Internet connections [5].

Greenstone's latest version, Greenstone3, is implemented using a decentralised agent-based design; making it scalable, flexible and extensible. The flexible design enables Greenstone to support distributed collections served from different machines [6].

The digital objects are stored on the file system. However, the metadata once imported into Greenstone is stored in a proprietary Greenstone Archive Format (GIF) that is not interoperable with other repository systems.

[3] http://www.eprints.org/

[4] http://fez.library.uq.edu.au/

[5] http://islandora.ca/

3 Hypotheses

The base of this research is composed of two working hypotheses that are as a result of grounding work conducted in the Digital Library Laboratory [11, 12]. The two hypotheses are:

- A formal simplistic abstract framework for Digital Library System design cab be derived.
- A Digital Library System architectural design based on a simple and minimalistic approach can be easily adopted and managed over time.

4 Research Questions

The core of this research will focus on investigating whether it is feasible to implement Digital Library Systems based on simple architectures. As such, this research will seek to provide a comprehensive solution to the following research question.

4.1 Is it feasible to implement Digital Library Systems based on simple architectures?

The main research question broadly seeks to investigate the viability of simple architectures. To this end, the following sub-questions will seek to provide solutions to the main research question.

4.1.1 How should simplicity for Digital Library storage and service architectures be defined?

There are core underlying principles common to Digital Library System architectural designs [1]. The derivation of the set of principles behind the simplyCT framework will be based on an initial set of grounded principles that are solely based on current experience with implementation of Digital Library Systems for developing countries. The initial set of principles are outlined below.

- Principle 1: Minimalism

  There should be a minimal use of software components --the architectural design should be made as simple as possible; adding complexity only where absolutely necessary.

- Principle 2: Do not impose on users

  Potential end users should not change their business rules in order to align themselves with architectural design requirements.

- Principle 3: Preservation by copying

  The preservation process should be simplified by making it possible for digital content with their associated metadata to be easily copied to another platform and be able to function with little or no configuration requirements.

- Principle 4: Web or no Web

  It should be possible for digital collections to function in the event that the environment within which they are deployed do not have the necessary infrastructure to host Web applications.

- Principle 5: Any metadata, any objects, any services

  It should be possible for any type of digital object, metadata or service to be integrated with a Digital Library System implemented based on the simplyCT framework.

- Principle 6: Everything is repeatable

  Each digital object will have a metadata file stored alongside it.

- Principle 7: Superimposed information

  The digital content that end users eventually access from Digital Library Systems is usually composed of sub-digital objects. As such, to facilitate management, access, organisation, retrieval and reuse of the sub-components, there should be a way for the superimposed information to be referenced from the individual digital objects.

- Principle 8: No API

  APIs should be explicitly avoided; where necessary, files should be read directly from storage.

This research will seek to demonstrate the how the principles behind the simplyCT framework will result in Digital Library Systems that are effective and efficient. It will be necessary to derive the principles through a detailed study and analysis of existing systems and details of how to do that are outlined in Section 5.1

4.1.2 What are the comparative advantages and disadvantages of simpler architectures to complex systems?

A number of Digital Library System architectures have been proposed over the past two-decades, ranging from those specifically designed to handle complex objects [8, 13] to those with an overall goal of creating and distributing collection archives [9, 15]. This research question will seek to investigate the advantages and disadvantages that simpler architectures have compared to well-established complex Digital Library architectures. This will among other aspects, involve establishing how well simple architectures support CMSes, portal collections and scalable collections.

5 Methodology

A clear demonstration that the simplyCT framework encompasses the core Digital Library principles is vital for the success of this research. This will be done systematically by first deriving the underlining Digital Library principles --through a detailed case study of existing Digital Library Systems. This will be followed by demonstrating how the principles map on to the simplyCT framework; this will be done with the aid of a prototype Digital Library System.

5.1 Case Study: Existing Digital Library Systems

A few popular existing systems will be studied and analysed in detail to provide an exhaustive set of core principles common to Digital Library Systems and also to provide a very clear understanding behind the theory of Digital Library Systems.

5.2 Extension of Prototype Digital Library System

The first implementation of the prototype Digital Library System based on the simplyCT framework will be done by two honours students --Miles Robinson will implement a curator interface and Stuart Hammer will work on an end user interface. It may however be necessary to implement further system functionalities that are outside the scope of their honours project.

5.3 Integration with Collections

A number of collections--at least two-- will be implemented and integrated with the prototype Digital Library System to assess the overall applicability of the tool.

Section 7 outlines the details of the evaluation process for this research.

6 Work Detail

6.1 Risk Analysis

The major risk factor identified is outlined in the following section.

6.1.1 Collaboration

The implementation of the prototype Digital Library System will be done in collaboration with two honours students. The overall Digital Library features may be far beyond what can be feasibly delivered within the time frame available for the honours project. Additionally, there is thus a high communication risk as the development team will be dispersed, and is composed of individuals who have never worked together before.

Classification	Impact	Likelihood	Mitigation Plan
Major	Critical	Medium	■ Regular group meeting during project scoping. ■ Agree on common development tools to be used. ■ Use of a distributed version control system (e.g. Git).

6.2 Timeline

The planned timeline for this research is presented in form of a Gantt chart, illustrated in Appendix I.

6.3 Resources

The following resources will be required to successfully carry out the project:

- A personal computer with access to the Internet and University network.
- Java Servlet Engine, Python Interpreter and XSLT Processor
- Sample digital collections (e.g. Bleek and Lloyd Collection)

6.4 Deliverables

The following deliverables are expected to be produced after successful completion of the project:

- Proof of concept - a set of principles underlining the theory behind the simplyCT framework.
- Working digital collections based on the simplyCT framework.
- Final Report - Thesis.

6.5 Milestones

The following is a chronologically ordered list of milestones the project is expected to yield:

- Research Proposal - July, 2011
- Static Digital Library Collection Prototype - August, 2011
- Case Study: Existing Digital Library Systems - October, 2011
- Background Chapter - November, 2011
- Extension of Prototype Digital Library System - January, 2012
- Collections Integration - February, 2012
- Proof of Concept Systems - March, 2012
- Final Evaluation - April, 2012
- Thesis - August, 2012

7 Evaluation

The overall research assessment will be based on the three axes of evaluation strategy coined by Tsakonas et al [3].

7.1 Efficiency

A series of controlled experiments will be set up to measure the efficiency of the framework and/or implementations based on the framework. The experiments will involve typical Digital Library tasks such as search, browse and maintenance related tasks such backup and recovery.

The results from the controlled experiments will be used to compare the framework with already existing solutions.

7.2 Effectiveness

A number of real life case study collections will be implemented to measure the effectiveness of the framework. This will be done by assessing the quality of services offered by the resulting Digital Library System.

7.3 Usefulness

A user study will be conducted to assess the quality of the interaction between end users and the system; and the relevance of a system implemented using the simplyCT framework. A typical task, to assess the usefulness of the system, may be development of third-party services.

The most likely candidates may be postgraduate students from the Digital Libraries laboratory research group, as they are expected to have had prior experience with existing Digital Library Systems.

A comparative study with already existing Digital Library System will be conducted to assess the overall applicability of the framework.

8 Anticipated Outcome

It is anticipated that the guiding principles of the simplyCT framework will result in the implementation of Digital Library Systems and/or services that will be easily adopted and fairly easy to maintain. This would especially be useful when implementing digital collections in environments characterised by limited resources, such as those used for cultural heritage collections in developing countries.

References

[1] Arms, W.Y. 1995. Key Concepts in the Architecture of the Digital Library. *D-Lib Magazine*. 1, 1 (Jul. 1995).

[2] Buchanan, G., Bainbridge, D., Don, K.J. and Witten, I.H. 2005. A new framework for building digital library collections. *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries - JCDL '05*. (2005), 23.

[3] Fuhr, N., Tsakonas, G., Aalberg, T., Agosti, M., Hansen, P., Kapidakis, S., Klas, C.-P., Kovács, L., Landoni, M., Micsik, A., Papatheodorou, C., Peters, C. and Sølvberg, I. 2007. Evaluation of digital libraries. *International Journal on Digital Libraries*. 8, 1 (Feb. 2007), 21-38.

[4] Gonçalves, M.A., Fox, E.A., Watson, L.T. and Kipp, N.A. 2004. Streams, structures, spaces, scenarios, societies (5s). *ACM Transactions on Information Systems*. 22, 2 (Apr. 2004), 270-312.

[5] Gutteridge, C. 2002. GNU EPrints 2 Overview. *11th Panhellenic Academic Libraries Conference* (2002).

[6] Kahn, R. and Wilensky, R. 2006. A framework for distributed digital object services. *International Journal on Digital Libraries*. 6, 2 (Mar. 2006), 115-123.

[7] Körber, N. and Suleman, H. 2008. Usability of Digital Repository Software : A Study of DSpace Installation and Configuration. *ICADL 08 Proceedings of the 11th International Conference on Asian Digital Libraries: Universal and Ubiquitous Access to Information* (Berlin / Heidelberg, 2008), 31-40.

[8] Lagoze, C., Payette, S., Shin, E. and Wilper, C. 2005. Fedora: an architecture for complex objects and their relationships. *International Journal on Digital Libraries*. 6, 2 (Dec. 2005), 124-138.

[9] Millington, P. and Nixon, W.J. 2006. EPrints 3 Pre-Launch Briefing. *Ariadne*.

[10] Pepe, A., Baron, T., Gracco, M., Le Meur, J.Y., Robinson, N., Simko, T. and Vesely, M. 2005. CERN Document Server Software: the integrated digital library. *ELPUB 2005 conference, Heverlee (Belgium)* (2005), 8–10.

[11] Suleman, H. 2007. Digital Libraries Without Databases: The Bleek and Lloyd Collection. *Proceedings of Research and Advanced Technology for Digital Libraries, 11th European Conference (ECDL 2007)* (Budapest, Hungary, 2007), 392-403.

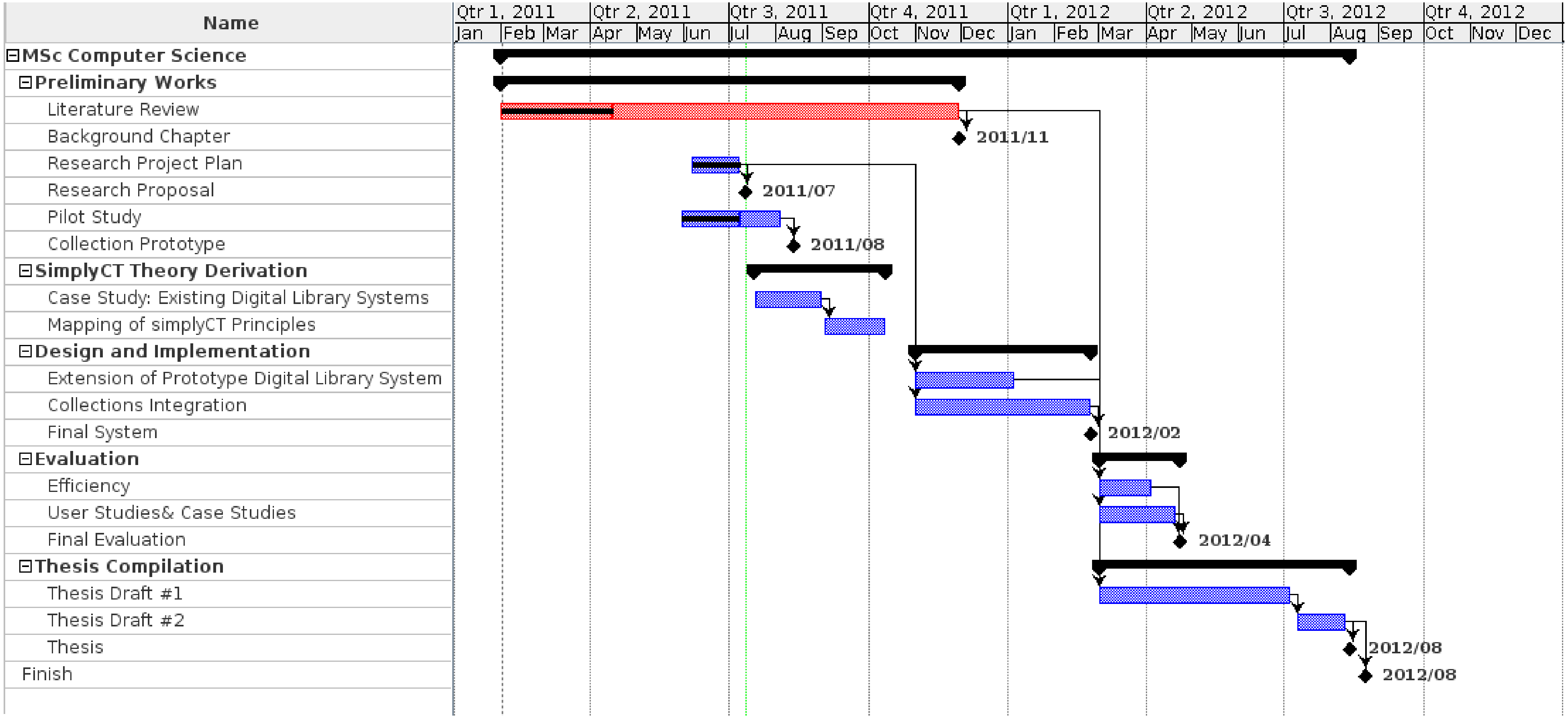
[12] Suleman, H., Bowes, M., Hirst, M. and Subrun, S. 2010. Hybrid online-offline digital collections. *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on - SAICSIT '10* (New York, New York, USA, 2010), 421-425.

[13] Tansley, R., Bass, M., Stuve, D., Branschofsky, M., Chudnov, D., McClellan, G. and Smith, M. 2003. The DSpace institutional digital repository system: current functionality. *2003 Joint Conference on Digital Libraries, 2003. Proceedings.* (2003), 87-97.

[14] Vesely, M., Baron, T., Le Meur, J.Y. and Simko, T. 2002. Creating open digital library using XML: implementation of OAi-PMH protocol at CERN. *Presented at Elpub*. (2002).

[15] Witten, I.H., Bainbridge, D. and Boddie, S.J. 2001. Greenstone: open-source digital library software with end-user collection building. *Online Information Review*. 25, 5 (2001), 288-298.

Appendix I: Gantt Chart Showing the Proposed Work Plan


Name
Qtr 1, 2011
Qtr 2, 2011
Qtr 3, 2011
Qtr 4, 2011
Qtr 1, 2012
Qtr 2, 2012
Qtr 3, 2012
Qtr 4, 2012
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
⊟MSc Computer Science
⊟Preliminary Works
Literature Review
Background Chapter
Research Project Plan
Research Proposal
Pilot Study
Collection Prototype
⊟SimplyCT Theory Derivation
Case Study: Existing Digital Library Systems
Mapping of simplyCT Principles
⊟Design and Implementation
Extension of Prototype Digital Library System
Collections Integration
Final System
⊟Evaluation
Efficiency
User Studies& Case Studies
Final Evaluation
⊟Thesis Compilation
Thesis Draft #1
Thesis Draft #2
Thesis
Finish
2011/11
2011/07
2011/08
2012/02
2012/04
2012/08
2012/08