SimplyCT

A Web-based Digital Repository System

Stuart Hammar Miles Robinson

University of Cape Town

Department of Computer Science

{shammar, mrobinson}@cs.uct.ac.za

1. Project Description and Motivation

1.1 Project Description

Custom Digital Repository Systems (DRSes) manage collections of data stored in predefined file stores. These file stores can range from complex databases and XML files to a simple file hierarchy. However, few DRSes make use of simple file stores to store collections.

The goal of SimplyCT is to develop a set of Web-based tools around a predefined simple data store. These tools will provide services such as search and browse to end-users. Furthermore, administrators will be provided with methods to manipulate digital collections and their metadata. These tools will be made accessible via a Web-based user interface.

Additionally, research will be concluded to determine whether the use of a hierarchical file-based data store creates any significant limitations in the development of a DRS.

1.2 Project Motivation

Current DRSes provide tools that use central databases and other data stores. These data stores are often considered complex and are not easily distributable. This exposes a research opportunity to investigate whether building a DRS around a simple file-based data store is feasible.

The CALJAX project attempted to create an offline DRS using Web 2.0 technologies [1]. This system requires only the use of the user's Web browser and does not require pre-installed software. CALJAX was built around providing services for and manipulating a predefined hierarchical data store.

The idea of CALJAX paves the way forward for the development of a DRS that takes advantage of the user's Web browser for accessing a simple data store. However, the focus of CALJAX was to develop an easily distributable and offline DRS.

The opportunity exists for an online DRS centred on a simple file-store to be explored. Thus, the aim of SimplyCT is to develop a general Web-based DRS solution that provides powerful end-user and collection management tools to explore and manipulate a predefined simple data store.

2. Problem Statement

2.1 Research Questions

Having been supplied with a file hierarchy as a digital data store, previous DRS solutions have been custom-built to make use of the file hierarchy for its users. The SimplyCT project is an attempt to create a general set of Web-based tools to access and manipulate this simple and lightweight file hierarchy. Three primary research questions need to be answered to determine the success of the project:

1. Will using a simple file hierarchy as a file store affect the overall usability of the system?
2. What is the impact of having a hierarchical file-based data store on the performance of the system?
3. Is it possible to create a user interface to a hierarchical file-store that is comparable to other DRSes?

Each research question will be evaluated to determine whether SimplyCT can be considered to be feasible or not.

2.2 Preliminary Requirements

As a starting point, the project members have devised a set of preliminary requirements that are thought to be beneficial to the end-users and curators of the proposed SimplyCT system. An overview of these preliminary features is discussed below.

Data Storage Format. The SimplyCT project is based on building a Web-based system around a simple data store – a set of hierarchical files. Any storage of data that is auxiliary to the primary data store (e.g. login information) will be stored in an economical and lightweight database system, such as SQLite.

End-User Services. From the literature survey conducted previously, it was noted that there are general end-user services that are assumed key for DRSes. Such services include search and browse functionality – as a result these services will be considered for SimplyCT. These services will provide a means for users to locate and identify information that they may be looking for. Additionally, other preliminary services that have been suggested by the project team include:

- *Administrator functionality*. This will allow the administrator to customise the look and feel of the webpage of the corresponding collection.
- *Annotations service*. Ideally, this will be adopted from various Web 2.0 services. This will allow registered users to post comments about a collection or digital object that they may be viewing.

Curator Services. The literature review also indicated that certain curator services are deemed key and standard with a DRS. Such services include the ability to manage and organise digital collections and their metadata. To facilitate this, the system will support adding, deleting and editing of digital objects. Other services that have been proposed by the team members include:

- *User management*. User management is a form of access control. Fundamentally, the curator will be able to specify which users are able to view the digital collection. The curator will also be able to revoke a user's permissions to view a collection if necessary.

The above-mentioned preliminary requirements merely provide an outline of expected system requirements and a further requirements gathering process will be conducted (discussed in *Section 3.1*).

3. Procedures and Methods

An iterative design and implementation process will be used throughout the SimplyCT project. The development process will consist of three iterations to be conducted:

- *Iteration 0.* The goal is to develop a rapid prototype of the system.
- *Iteration 1.* The goal is to develop a more complete second prototype.
- *Iteration 2.* The goal is to have developed a final working system.

Each of the iterations will consist of three phases: the design phase, the implementation phase and the evaluation phase. An overview of each of these three phases is provided below.

3.1 Design Phase

The design phase for each of the iterations will help to refine the user requirements to be forwarded to the implementation phase.

The design phase of *Iteration 0* is arguably the most important phase of the project. In this phase the project team will gather information as to the system requirements of a DRS. Initially, the team members will need to conduct further background research into DRSes to determine any key features that may not have been previously known. Subsequently, experienced digital repository users will need to be consulted. These users will be interviewed in a focus group. The focus group will help to provide the team members with further knowledge as to what is to be expected from a DRS and what can be applied to SimplyCT.

The design process for *Iterations 1* and *2* will be somewhat less in-depth as in *Iteration 0* because further user requirements will not need to be acquired. In these two iterations, the design process will involve applying the results from the evaluation process of the previous iteration and ensuring that any changes suggested and problems noted are implemented and rectified for the next prototype.

3.2 Implementation Phase

The implementation phase will comprise of developing the proposed system to incorporate all of the features uncovered in the design phase.

In *Iteration 0* the team members will develop an initial rapid prototype to have the basic structure of the Web service complete. When developing the first prototype the user requirements discovered in the design process will need to be adhered to. The features of the service will not have to be fully functioning or optimised, but a good idea of the direction of the system will be constructed.

Iteration 1 will again involve using the new design considerations explored in the design process to add to and improve the second prototype to be developed. In this instance full feature functionality should be available. However, the functions may not be fully optimised. It is expected that after this implementation only optimisations and minor adjustments to the system will be necessary.

The implementation process of the final iteration will ensure that the system has all of its relevant functions completed and optimised. This will ensure that a full set of evaluations can be conducted on the system.

3.3 Evaluation Phase

The iterations will include an extensive evaluation phase. The evaluation phase will help to uncover any errors or inconsistencies in the system that may not meet the suggested user requirements or hinder system performance.

The evaluation method of *Iteration 0* will undoubtedly be “quick and dirty”. This primary evaluation will be to determine that the users’ requirements have been met and that the system is accessible and usable. Therefore, only a small set of users will be used to conduct usability testing. After the user testing is conducted, a short feedback session will be held to uncover any incongruities in expected system usage. Evaluations for system efficiency and comparative performance will be irrelevant at this point.

Iteration 1 will again involve user testing to evaluate the usability and accessibility of the system. Once more, a set of tasks will be set out for the user to complete. Upon completion of the tasks a feedback session will be held to discuss any issues that may have surfaced. In this iteration, performance-based testing will be conducted on the system to see how the system performs with various amounts of data in the simple file store. Due to the system still being in its prototype phase, it does not seem feasible to conduct a comparative evaluation of systems yet.

The final evaluation will ensure the determination of the answers to the three proposed research questions of the project. A significant amount of time will be set aside to complete this evaluation set. A final usability test will be conducted for each section of the system. This will be followed by a rigorous feedback session in which the system will be critically analysed. This will determine whether the system behaves as the user expects and is sufficiently usable. Performance-based testing will be conducted to determine if the predefined file system hinders the efficiency of the system. Lastly, a comparative evaluation of SimplyCT’s user interface will be conducted to determine whether or not it is comparable to the interfaces of other DRSes.

4. Ethical, Professional and Legal Issues

No ethical, professional or legal issues are anticipated to arise in the project. Should ethical clearance be required when conducting user testing, appropriate measures will be taken. All of the software to be used in development is open source and free of charge.

5. Related Work

Several other digital library systems have been developed. Some of these systems will now be briefly analysed.

Greenstone digital library software is an open source system that allows digital collections to be constructed [7, 8]. These digital collections can then be exported onto static media. This allows the repository to be viewed offline. However, a Web server is still required in order for Greenstone to run. Although Greenstone is designed to be accessed over the Web, basic software installation is still required. This is due to the complex file storage system that Greenstone uses. As a result of the installation requirement, the portability of Greenstone is not as good as some of the other systems available.

The Bleek and Lloyd Collection uses static pages. This allows the collection to be exported, like Greenstone, to static media. However, unlike Greenstone, the Bleek and Lloyd Collection does not require a Web server [5]. The Bleek and Lloyd Collection makes use of XML-based formats and thus has a high level of portability. There are, however, concerns over the scalability of the approach used in the Bleek and Lloyd collection. In addition, the approach used cannot be easily extended to manage other collections.

The CALJAX project made use of binary files to store a digital object [4]. Coupled with each digital object, was a metadata file that followed an XML schema. This storage system lends itself well to portability. The results of CALJAX indicated that it is possible to use a simple file structure in a database.

Acumen is an example of another lightweight digital repository system [2]. Acumen makes use of a file structure to store digital objects. Metadata associated with a digital object is entered into a spreadsheet. XML files containing an object's metadata are then generated from this spreadsheet. Acumen does require installation. However, it can be installed on any Operating System, therefore demonstrating portability.

Widely used digital repository systems, such as DSpace and Fedora, often make use of complex databases to facilitate storage [3, 6]. The primary aim of this project is to make use of a simple file storage structure. Therefore, these systems will not be considered for file storage. However, they may provide insight into the creation of user and curator services.

The Bleek and Lloyd Collection, CALJAX and Acumen are of particular interest to this project, as they have addressed similar areas of concern regarding file storage. Greenstone is also of interest as it does lend itself to high portability. However, its file storage method is different to that defined for this project.

6. Anticipated Outcomes

6.1 System

On completion, the project is expected to have delivered a Web-based DRS centred on a pre-determined file hierarchy. The system will have shown that it can provide an effective user experience by providing good usability. Additionally, the use of the hierarchical data store must not hinder the efficiency of the system. It will further be shown that the SimplyCT user interface is comparable to current DRSes.

SimplyCT will support browse, search and management functionality as standard, with additional end-user and curator services being developed. Such end-user services may include an annotations system and administrator customisation. On the management side, the system will incorporate an access control model. This will allow administrators to manage users and their access to the system. Curators will manage any assigned collections on the system and will allow users to access the collection information.

The major system challenges that are anticipated will be regarding the efficiency of the system when the file hierarchy expands to house more data. This will determine whether the expected outcome of efficiency can be maintained.

6.2 Expected Impact

The success of the project will determine the impact it has on the digital library community. Provided the project is successfully completed, the impact will be demonstrated in the context of new collections and communities being created. Organisations such as the Centre for Curating the Archive (at the University of Cape Town), the Indira Gandhi National Centre for the Arts (in New Delhi) and the District Six Museum may be interested in implementing such a system.

The success of a Web-based DRS centred on a simple data store will illustrate that such a system is feasible. It will also display that such a system can be created to run efficiently and not detract from the usability of the system. This will demonstrate that complex data structures may not be necessary for digital repository systems.

6.3 Key Success Factors

The success of the system will greatly be determined by the evaluations conducted on it. Evaluations addressing the three research questions will be carried out in each of the development iterations. To determine whether the system has succeeded, the evaluation results need to show:

- That the usability of the system and the user experience is not hindered by the simplicity of the data store.
- That the efficiency of the system is acceptable for large amounts of data in the data store.
- That the user interface of SimplyCT is comparable to those of other DRSes.

If the evaluation of the system does produce these results, the SimplyCT project will be considered a success. However, even if negative results are returned, they can still be considered as useful research into the use of simple data stores.

7. Project Plan

7.1 Risks

Below is a description of the perceived risks that may arise in the project, along with their impacts[*], likelihoods[†] and mitigation plans:

The final system does not work (Likelihood: 2; Impact: 8)

Mitigation Strategy: Develop according to the proposed timeline to ensure that working versions have been developed in time for each evaluation.

The final system does not include all original functionality (Likelihood: 7; Impact 3)

Mitigation Strategy: Use information gathered from initial design phase to determine which functionality is most important. These functions should be developed first so that a useful system is developed regardless of whether or not all functionality has been implemented.

Member does not complete their section of work (Likelihood: 2; Impact: 2)

Mitigation Strategy: A clear division of work will allow one member of the group to hand in a working system without being affected by the other member's contribution.

Design approach taken is infeasible (Likelihood: 2; Impact: 8)

[*] Impact is measured out of 10; 1 is "low impact" and 10 is "high impact".

[†] Likelihood is measured out of 10; 1 is "low chance of occurrence" and 10 is "high chance of occurrence".

Mitigation Strategy: Research has indicated that the file storage system can be implemented. Should problems occur, the previous solutions can be analysed.

System failure causes loss of data (Likelihood: 3; Impact: 4)

Mitigation Strategy: Perform regular backups, so that if there is a loss of data, previous versions can be used.

Supervisor leaves project (Likelihood: 2; Impact: 7)

Mitigation Strategy: Ensure that regular communication with the supervisor occurs so that if there is a possibility of the supervisor leaving, the necessary actions can be taken.

Uneven distribution of work (Likelihood: 1; Impact: 6)

Mitigation Strategy: Get the distribution of work approved by the project supervisor before the first iteration.

Project members do not get along (Likelihood: 2; Impact: 6)

Mitigation Strategy: Divide the work in such a way that minimal interaction between members is required. Any conflicts of interest between them should be discussed with the supervisor so that an appropriate solution can be found.

7.2 Timeline

- See Figure 1 in Appendix A for Gantt chart.
- See Table 1 in Appendix A for timeline.

7.3 Resources Required

Equipment

- A Web server
- Backup storage
- Workstations for each member of the group to work on

Software

All software required is open-source, free and readily available:

- SQLite
- SQLite connection libraries

7.4 Deliverables

The expected deliverables for this project include a detailed report written by each of the group members documenting their section of the project – end-user and curator systems. The programming code for each of these sections will be submitted. A poster for the project will need to be created. Lastly, a reflective essay on the project will be completed.

7.5 Milestones

- See Table 1 in Appendix A for the project milestones.

7.6 Work Allocation

Primarily, the SimplyCT project is split in two: the end-user services and the management services. These two sets of services and interfaces are evenly split between the two members.

- Stuart Hammar will be dealing with the end-user services and interface. He will conduct his three iterations and their corresponding phases separately to Miles.
- Miles Robinson will be managing the administrative services and its interface. Miles will conduct his three iterations and phases independently of Stuart.

Due to the work being completely separated, if one of the group members happens to leave the project, the other will not be reliant on that user's section. In other words, the other member's work will be unaffected.

8. References

[1] Bowes, M. CALJAX : An In-Browser Digital Repository System. *Honours Project Report*, Department of Computer Science, University of Cape Town (2009). http://people.cs.uct.ac.za/~mbowes/honsproj/resources/files/report.bwsmar002.pdf

[2] Leowald, T. and DeRiddler, J. Metadata In, Library Out. A Simple, Robust Digital Library System. *Code4Lib Journal*, 10, 2010. http://journal.code4lib.org/articles/3107.

[3] Staples, T., Wayland, R. and Payette, S. The Fedora Project: An Open-source Digital Object Repository Management System. *D-Lib Magazine*. *9* (4). 2003.

[4] Subrun, S. CALJAX: Management System of an AJAX based Digital Repository. *Honours Project Report*, Department of Computer Science, University of Cape Town (2009). http://people.cs.uct.ac.za/~mbowes/honsproj/resources/files/report.sbrsur002.pdf

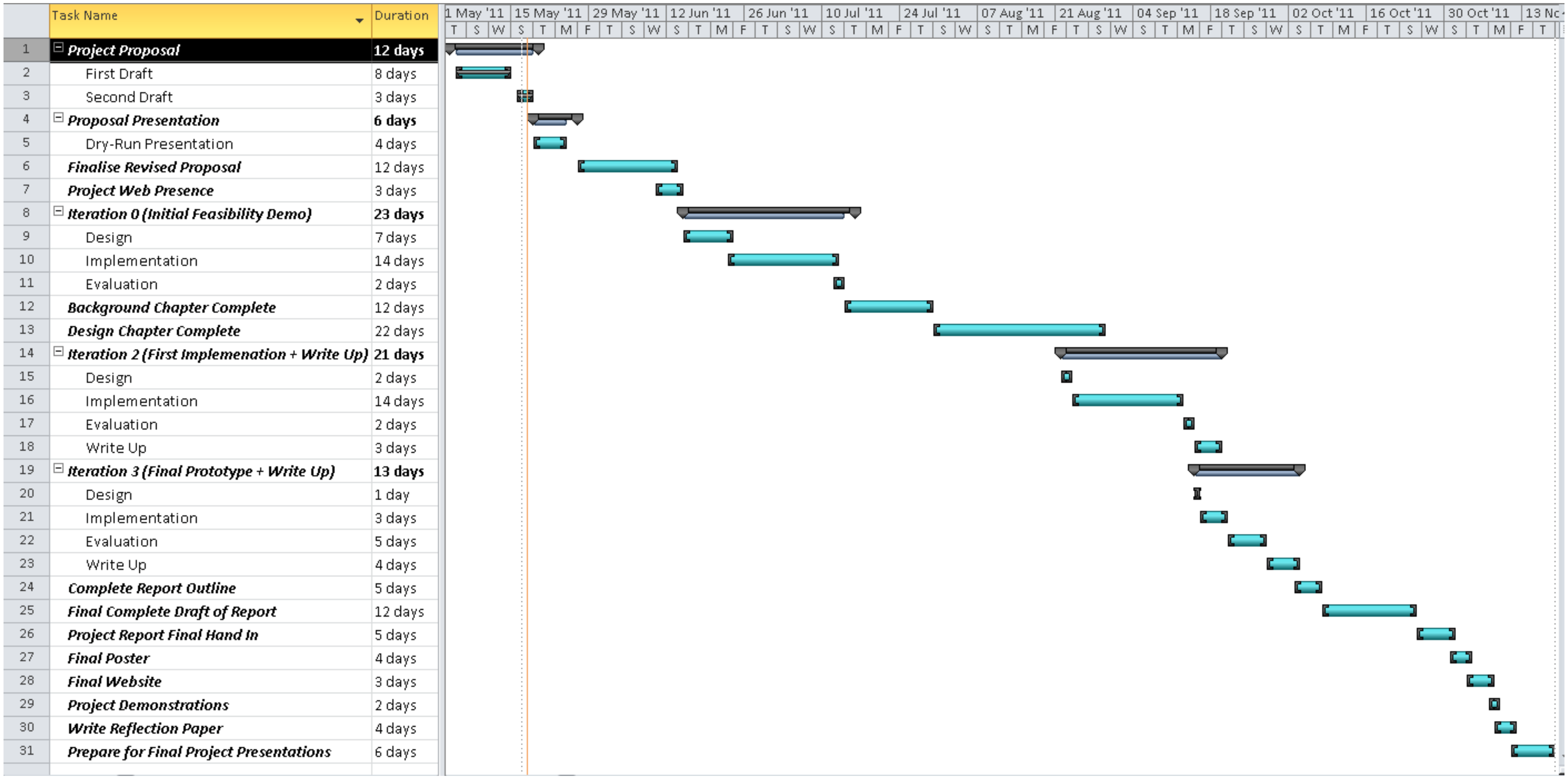
[5] Suleman, H. Digital Libraries Without Databases: The Bleek and Lloyd Collection. *Proceedings of Research and Advanced Technology for Digital Libraries, 11th European Conference* (ECDL 2007), 392-403, 16-19 September 2007.

[6] Tansley, R., Bass, M., Stuve, D., et al. The DSpace Institutional Digital Repository System: Current Functionality. *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital Libraries*, IEEE Computer Society (2003), 87–97.

[7] Witten, I.H., Bainbridge, D., and Boddie, S.J. Greenstone: Open-Source Digital Library Software with End-User Collection Building. *Online Information Review 25*, 5 (2001), 288-298.

[8] Witten, I.H., Boddie, S.J., Bainbridge, D., and McNab, R.J. Greenstone: A Comprehensive Open-Source Digital Library Software System. *Proceedings of the Fifth ACM Conference on Digital Libraries*, ACM (2000), 113–121.

Appendix A



	Task Name	Duration
1	***Project Proposal***	**12 days**
2	First Draft	8 days
3	Second Draft	3 days
4	***Proposal Presentation***	**6 days**
5	Dry-Run Presentation	4 days
6	***Finalise Revised Proposal***	12 days
7	***Project Web Presence***	3 days
8	***Iteration 0 (Initial Feasibility Demo)***	**23 days**
9	Design	7 days
10	Implementation	14 days
11	Evaluation	2 days
12	***Background Chapter Complete***	12 days
13	***Design Chapter Complete***	22 days
14	***Iteration 2 (First Implemenation + Write Up)***	**21 days**
15	Design	2 days
16	Implementation	14 days
17	Evaluation	2 days
18	Write Up	3 days
19	***Iteration 3 (Final Prototype + Write Up)***	**13 days**
20	Design	1 day
21	Implementation	3 days
22	Evaluation	5 days
23	Write Up	4 days
24	***Complete Report Outline***	5 days
25	***Final Complete Draft of Report***	12 days
26	***Project Report Final Hand In***	5 days
27	***Final Poster***	4 days
28	***Final Website***	3 days
29	***Project Demonstrations***	2 days
30	***Write Reflection Paper***	4 days
31	***Prepare for Final Project Presentations***	6 days

Figure 1. Gantt chart for the project.

Table 1. Timeline indicating task start and finish times of tasks and milestones. Milestones appear in italics and bold.

Task Name	Duration	Start	Finish
Project Proposal	***12 days***	***04 May 2011***	***19 May 2011***
First Draft	8 days	05 May 2011	14 May 2011
Second Draft	3 days	16 May 2011	18 May 2011
Proposal Presentation	***6 days***	***19 May 2011***	***26 May 2011***
Dry-Run Presentation	4 days	19 May 2011	24 May 2011
Finalise Revised Proposal	12 days	27 May 2011	13 June 2011
Project Web Presence	3 days	10 June 2011	14 June 2011
Iteration 0 (Initial Feasibility Demo)	***23 days***	***15 June 2011***	***15 July 2011***
Design	7 days	15 June 2011	23 June 2011
Implementation	14 days	23 June 2011	12 July 2011
Evaluation	2 days	12 July 2011	13 July 2011
Background Chapter Complete	***12 days***	***14 July 2011***	***29 July 2011***
Design Chapter Complete	***22 days***	***30 July 2011***	***29 August 2011***
Iteration 2 (First Implemenation + Write Up)	***21 days***	***22 August 2011***	***19 September 2011***
Design	2 days	22 August 2011	23 August 2011
Implementation	14 days	24 August 2011	12 September 2011
Evaluation	2 days	13 September 2011	14 September 2011
Write Up	3 days	15 September 2011	19 September 2011
Iteration 3 (Final Prototype + Write Up)	***13 days***	***15 September 2011***	***03 October 2011***
Design	1 day	15 September 2011	15 September 2011
Implementation	3 days	16 September 2011	20 September 2011
Evaluation	5 days	21 September 2011	27 September 2011
Write Up	4 days	28 September 2011	03 October 2011
Complete Report Outline	***5 days***	***03 October 2011***	***07 October 2011***
Final Complete Draft of Report	***12 days***	***08 October 2011***	***24 October 2011***
Project Report Final Hand In	***5 days***	***25 October 2011***	***31 October 2011***
Final Poster	***4 days***	***31 October 2011***	***03 November 2011***
Final Website	***3 days***	***03 November 2011***	***07 November 2011***
Project Demonstrations	***2 days***	***07 November 2011***	***08 November 2011***
Write Reflection Paper	***4 days***	***08 November 2011***	***11 November 2011***
Prepare for Final Project Presentations	***6 days***	***11 November 2011***	***18 November 2011***