CALJAX
AJAX-based in-Browser Digital Repository system

Marc Bowes, Matthew Hirst and Suraj Subrun

May 2009

1 Project Description

This project aims to implement a system to browse, search and manage a digital collection. This system should be lightweight, extensible and distributable, with a minimal software footprint without sacrificing functionality.

The system should feature basic browsing, searching and updating. The update functionality should include the ability to update a central repository.

2 Problem Statement

2.1 Research question

Digital Repository systems are typically heavyweight, requiring a Web server in both online and offline forms. This project will attempt to overcome this through the use of Asynchronous Javascript and XML (AJAX) in a Web browser. The research question asks whether this approach is feasible and able to provide at least the basic functionality expected of a Digital Repository system.

2.2 Motivation

An offline Digital Repository system is desirable for distributing content in environments which have limited to no Internet connectivity. Greenstone, a popular distributable Digital Repository system, distributes 50 000 copies of it's software a year [1]. However, it is not always possible to install software in the target environment to run a system such as Greenstone. This project aims to plug this gap by minimising software requirements, using only the typical suite of installed software. The Bleek and Lloyd Collection provides a sway of distributing a digital collection, allowing users to browse and search the collection using only their Web browser [2]. However, this system is specific for their content and cannot be generalised to fit an arbitrary collection of data. This project aims at a more general solution - that is, it should be applicable to a wide variety of content types and collections.

3 Work Allocation

Work in this project has been divided up by functionality. The functions themselves divide up into three main categories making it an ideal way to split the project between the three group members.

- Matthew Hirst has been assigned the task of browsing the digital repository.
- Marc Bowes has been assigned the task of searching the digital repository.
- Suraj Subrun has been assigned the task of managing and updating the repository.

If one member of the project becomes unavailable, or is unable to complete their section, the only consequence is that the system as a whole loses that functionality - the other members are generally unaffected.

Figure 1 demonstrates how the key components of the system interact. Browsing and searching both query the repository. Updating and management

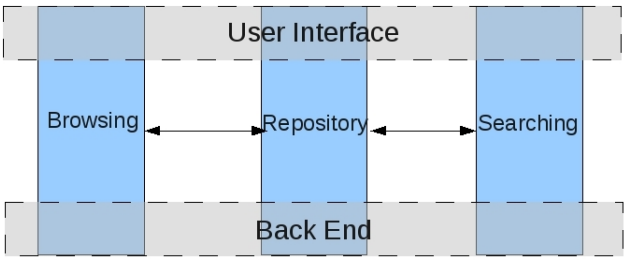

Figure 1: How the various sections interface

modifies the repository, which affects both browsing and searching.

4 Procedures and Methods

4.1 Browsing

Description

The task of building a system to browse a digital repository has been solved in the past by using files' metadata to link to them. In this solution to the problem of browsing a digital repository, linking to files using the metadata will similarly be used.

However because this system is using AJAX in its implementation, it is not possible to connect to a database or filestore directly due to security restrictions[5]. Methods that do not directly browse the filestore from AJAX need to be used. One method to work around this is to use the Javascript from AJAX to load hyperlinks embedded in the page.

This approach requires pre-processing the data and embedding links to the information within the HTML page. This approach was used by a lightweight AJAX wiki tool known as TiddlyWiki [5].

In order to accomplish this, an external program will need to be implemented to search through the objects in the digital repository and generate a hierarchy to browse through. This way the external program will generate an HTML page with all the links embedded, as well as the static AJAX code needed to load the links. The AJAX engine embedded in the webpage will then load the data and browse the repository through the embedded links.

Evaluation

To test the system, two attributes need to be examined. Firstly the compatibility of the system will need to be tested. This will be accomplished by testing the system on different operating systems and browsers. The aim of the project is to be a highly compatible, distributable system, and as such needs to work on a variety of operating systems and browsers.

The second attribute is the usability of the system. This will need to be tested through user-based tests. A group of users will be asked to use the system and then asked to provide feedback.

4.2 Searching

Description

Searching the Digital Repository requires that preprocessing be done in order to quickly perform queries. The preprocessor needs to build inverted files, mapping words to the documents in which they appear. These index files are then used by AJAX code in the Web browser to perform a search.

The preprocessor will be implemented in a Java Applet, so that updates can be done in-browser. However, performance could be poor, and it might be worthwhile implementing an external version purely for generating collections (with the Java Applet intended for incremental, rather than full, updates). The preprocessor will need to scan all of the documents and build index files. Other techniques, such as clustering, could be implemented in order to improve the quality of the search results. Since the core project goal is to prove the viability of the AJAX-only approach, emphasis will be placed on compatibility and speed, rather than information retrieval techniques.

Evaluation

The success of the search functionality will be measured by user tests. The aim of the test will be to determine if the user is able to retrieve certain documents while maintaining efficiency in time and space.

The system will be tested across multiple platforms, to ensure that performance (and compatibility) is good in varied environments. Tests should also

be done across Web browsers, as each browser implements their own Javascript engine which may perform differently (or not at all).

4.3 Management and central repository updating

Description

The system implemented will primarily consist of portable collections which will be managed individually offline. However, there will be the option of updating an online central repository with content from private user collections. It is expected that there will be no major differences between versions of the software that act as central servers and those used as portable collections.

Management of the software will be divided into two major components: a core layer and a storage layer. The core layer will implement the management and access subsystems and will provide the operations needed for viewing, creating, modifying, deleting and maintaining digital objects in the repository. It might also feature validation and integrity controls on the data. The storage layer will be responsible for storing and retrieving and removing the data from the repository. The main reason for this task subdivision is that the two layers have different environment requirements and can be implemented independently of each other. The storage layer could then be optimised to work under different platforms and environment.

The core layer can be implemented using AJAX which is a standard feature of most current Web browsers. The core layer might additionally feature access control and administration and support for standard metadata formats.

The storage layer however will require the use of additional technologies since Javascript does not allow file access. Using Java is a viable option since it is in widespread use and usually does not require any additional software installations. Key design decisions about the storage layer involve the way in which files will be stored locally. The use of a database system to store digital objects might provide the fastest and most effective solution to storing data but will compromise portability. Dynamically indexed XML files will also be considered as a means of storing the digital repositories data.

Central repository updating will be carried out using Web 2.0 features for the interaction between the central repository server and the clients. Web 2.0 technology is prevalent in most standard browsers and will be ideal for such a feature.

Evaluation

One way of evaluating the system would be to test it under different conditions and systems. Its efficacy will then be assessed from its ability to work on a large number of platforms. User evaluations of the system might be conducted to test its features and to assess its practicability as a lightweight alternative to the systems in use presently.

5 Ethical, Professional and Legal Issues

There are no forseeable ethical or legal issues with respect to either implementation or exploration of the proposed project. Any user testing will consider all relevant ethics. No licensing requirements are forseen, as all anticipated creation tools are free to use.

6 Related Work

Greenstone is a Digital Repository system which makes it possible to export a digital collection for offline viewing [1]. The resulting system is static, but requires a Web server in order to run [3]. Greenstone collections can be updated, but this requires a rebuild which can take up to two days [3].

The Bleek and Lloyd Collection also exports a digital collection as static pages, but does not require a Web server [2]. However, this system is specific for it's data.

DSpace and EPrints are both online solutions with software dependencies on Java and Perl respectively, as well as a database store such as Oracle or PostgreSQL [4].

Of the four systems, three of them require software installation. While the Bleek and Lloyd Collection meets most of the project requirements, it cannot be extended to manage other collections. Additionally, it does not provide functionality to manage the content - once generated the system is completely static.

Techniques used by Greenstone and the Bleek and Lloyd Collection are of most interest to this project as a combination of the systems would meet most project requirements (Greenstone's extensibility and the Bleek and Lloyd Collection's software-independence).

7 Anticipated Outcomes

7.1 System

The Digital Repository system implemented should support the main features expected of a standard digital repository software system. It should be browsable, searchable and it has to provide management item functionality. The system should also feature central repository update facilities. In contrast to most current systems in use, it should be able to operate in the absence of an Internet connection and it should have minimal installation and operation requirements. Optimally there should be no installation required and it should support various system architectures.

The design challenges faced will concern the methods used to browse, search and manage the data. For storage, the several options available include using underlying databases or custom XML files. The security restrictions on Javascript do not allow local file access so the system might require the use of additional technologies such as an embedded Java Applet. These would have to be chosen carefully because they are likely to affect the portability of the software.

7.2 Expected impact

If the project is successful, it has potential to have a profound impact or at least pave the way for one. A successful project would enable low cost content distribution, which has excellent educational value. Farming, medical or cultural data can be compiled onto low cost media (such as a CD or DVD- ROM) and distributed to just about any computer. Almost all operating systems have a Web browser, meaning that almost anybody can use the collection.

The project would also be invaluable for research. Due to the Web nature of the system, one is able to host both offline and online versions of a repository. A researcher could use the online version as any other Digital Repository system, or could request a compiled version, which is very useful when the researcher might be unable to access the Internet. The researcher would also be able to update the repository - either his own offline version, or the central repository - should the need arise.

7.3 Key success factors

The success of this project is based on evaluating each primary feature, where the features are browsing, searching, updating the collection as well as the portability of the system. If any of these features are compromised, it will most likely jeopardise the project.

While it is not imperetive to have excellent browsing and searching, it is important that the features do not detract from the overall experience. Features will be evaluated through user studies, to verify that the project is usable as a Digital Repository system.

System portability testing is a matter of installing the system on as many varied machines as possible. While it is unlikely that absolute portability will be achieved, it is expected that the system be usable in any AJAX-powered Web browser. There may be some hitches with regards to Java Applet support, and these will be tested for.

8 Project Plan

8.1 Risks

- See Table 1 in Appendix A.

8.2 Timeline

- See Table 2 in Appendix A for timeline.

- See Figure 2 for Gantt chart.

8.3 Resources required

Equipment

- Workstations for all members of the group to work on for the duration of the project
- Backup space
- Web server

Software

- Asynchronous Javascript and XML
- XML processing libraries

8.4 Deliverables

The deliverables expected to be produced by this project are a report on each aspect of the AJAX repository system, as well as a functional prototype of the system itself. The reports will include browsing, searching and managing.

8.5 Milestones

- See Table 3 in Appendix A.

References

[1] Witten, I.H., Bainbridge, D., Boddie, S.J., "Power to the People: End-user Building of Digital Library Collections", Proceedings of the 1st ACM/IEEE-CS joint conference on Digital Libraries, p.94-103, Virginia, United States, 2001.

[2] Suleman, H., "Digital Libraries Without Databases: The Bleek and Lloyd Collection", Proceedings Research and Advanced Technology for Digital Libraries, 11th European Conference, p.392-403, 2007.

[3] Witten, I.H., McNab, R.J., Boddie, S.J., Bainbridge, D., "Greenstone: A Comprehensive Open-Source Digital Library Software System", Proceedings of the fifth ACM conference on Digital libraries, p.113-121, June 02-07, 2000, San Antonio, Texas, United States.

[4] Kim, J., 2006, "Finding Documents in a Digital Institutional Repository: DSpace and EPrints", Proceedings 68th Annual Meeting of the American Society for Information Science and Technology, Charlotte, North Carolina: American Society for Information Science and Technology.

[5] Rees, M.J., 2006, "Ultra Lightweight Web Applications: A Single-Page Wiki employing a Partial Ajax Solution". [Online]. Available: http://ausweb.scu.edu.au/aw06/papers/refereed/rees/paper.html. [May. 5, 2009].

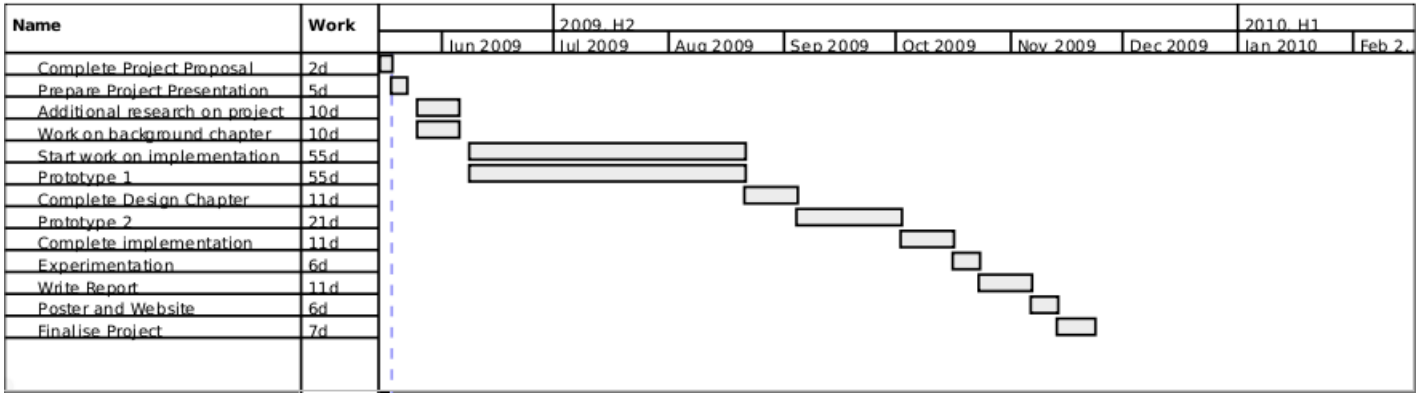

Figure 2: Gantt Chart

TABLE 1: RISKS

Risk ID	Risk	Probability of Risk	Impact of Risk
1	Group member leaves project group.	Medium	Medium
2	Group member does not complete their section of work.	Low	Medium
3	Software used in approach does not work. E.g. AJAX is not applicable to the problem.	Low	High
4	Loss of work due to system crashes or other technical failure.	Low	High
5	Scope of project is too small or too large.	Medium	Medium

TABLE2: Timeline

Further research into topic and work on project proposal.	10/05 - 15/05
Complete Project Proposal	15/05 - 18/05
Prepare project presentation	18/05 - 22/05
Additional research on project Work on background chapter	25/05 - 5/06
Start work on implementation Prototype 1	8/06 - 21/08
Complete Design Chapter	21/08 - 4/09
Prototype 2	4/09 - 2/10
Complete implementation of system	2/10 - 16/10
Experimentation	16/10 - 23/10
Write Report	23/10 - 6/11
Poster Website	6/11 - 13/11
Finalize Project	13/11 - 23/11

TABLE 3: MILESTONES

Project Proposal Documentation 1st Draft to project supervisor.	15/05/2009
Project Proposal Hand In	18/05/2009
Project Proposal Presentation	22/05/2009
Complete Background Chapter	31/07/2009
Prototype Demo	21/08/2009
Finish Design Chapter	4/09/2009
First Implementation/Experiment/Performance Test + Writeup	2/10/2009
Final Prototype/Experiment/Performance Test + Writeup	16/10/2009
Chapters on Implementation and Testing. Final implementation (optimised, etc.) completed, testing completed.	23/10/2009
First Draft of Report	30/10/2009
Project Report Final Handin	6/11/2009
Poster	13/11/2009
Website	13/11/2009
Self Reflection	13/11/2009
Final Project Presentation	23/11/2009 - 24/11/2009