Flexing Digital Library Systems

Hussein Suleman, Kevin Feng, Siyabonga Mhlongo, Muammar Omar

Department of Computer Science, University of Cape Town
Private Bag, Rondebosch, 7701, South Africa
{hussein, ffeng, smhlongo, momar}@cs.uct.ac.za

Abstract. Digital library systems with monolithic architectures are rapidly facing extinction as the discipline adopts new practices in software engineering, such as component-based architectures and Web Services. Past projects have attempted to demonstrate and justify the use of components through the construction of systems such as NCSTRL and SchoolNet. This paper describes current work to push the boundaries of digital library research and investigate a range of projects made feasible by the availability of suitable components. These projects include: the ability to assemble component-based digital libraries using a visual interface; the design of customisable user interfaces and workflows; the packaging and installation of systems based on formal descriptions; and the shift to a component farm for cluster-like scalability. Each of these sub-projects makes a potential individual contribution to research in architectures, while sharing a common underlying framework. Together, all of these projects support the hypothesis that a consistent component architecture and suite of components can provide the basis for advanced research into flexible digital library architectures.

1 Introduction

It is fast becoming recognised that current models in software engineering need to be integrated and applied to digital libraries. Most important among these models are the pivotal role of simplicity of design and the construction of larger systems from components [5, 3].

Some component frameworks have emerged in recent years to attempt to model systems as networks of loosely connected components instead of the traditional monolithic model. The Open Digital Library project (ODL) [7] generalised the well-understood syntax and semantics of the OAI-PMH to support general inter-component communication. This generalisation was then used as the basis for designing a suite of simple protocols to support search engines, category-based browsing, recommendation systems, annotation engines and other typical services expected by users of a digital library. Components, corresponding to each of these protocols, were created and connected together to test the performance of such systems and the ability of the model to elaborate various different types of digital library systems. The results of such tests [8] showed that the model has much promise. At the same time, feedback from users and developers has indicated that while simplicity of the individual components is useful, much work

still needs to be done in order to simplify the process of going from a set of components to a fully-fledged and seamless digital library.

Concurrent with the development of the ODL model, similar efforts were underway on the OpenDLib project [2]. The aims of both projects are similar, but the approach differs in that OpenDLib uses a transport layer that is composed of custom protocols layered over SOAP. Lessons learnt from both projects can ultimately lead to the creation of a standardised component model.

These models were proposed to support flexible digital libraries, and simplicity of components has proven to be popular. The natural next step is to investigate higher level techniques to support the creation of complete digital libraries from components in a simple and flexible manner. This paper thus provides an overview of a series of experiments conducted with components in the ODL family, to demonstrate higher level functionality in creating systems, while discovering some of the requirements for component frameworks in order to support such higher level functionality. Details of the ODL framework are omitted for brevity but can be found in referenced publications [7, 8].

2 Experiments

The main aim of these experiments was to investigate techniques, models and tools for constructing flexible digital libraries based on simple components arranged into a network of services. To this end, a number of questions were asked and tackled relatively independently:

- How do we create visual interfaces to compose components into complete systems and how do we specify the connections between components?
- How can the user interface and workflow be designed and specified to create a customisable front-end to back-end components?
- How can systems made up of components be packaged for use at remote sites, maintaining flexibility while promoting rapid deployment?
- Since these components are largely independent of one another, can they be run on a cluster of computers instead of a single system, thereby gaining the advantages of robustness and scalability?

2.1 Visual Component Composition

The BLOX system [4] was developed to demonstrate that a digital library could be constructed using a visual IDE, similar to those used in conjunction with popular programming languages. A suite of services corresponding to the abstract model of a digital library could be created and clicking on a "Publish" button instantly created and configured all component instances on a live server!

Extensive testing was conducted on the usability and utility of BLOX as compared to older methods of manually installing and configuring digital library components. The overwhelming results of the evaluation indicate that users would far rather prefer a graphical interface because of the familiarity and

flexibility that it affords. This is not atypical but confirms that digital library systems and Web-based information management systems in general need to move towards simpler and more customisable configuration procedures.

A major contribution of this study was the development of a simple descriptive language (similar to the 5SL project [6], but simpler and more specific to ODL) for specifying the interconnections among components, and a standardised interface for the remote management of components and component instances (creating, listing, editing, listing types, etc.).

2.2 Interface Customisation

In addition to customising the collection of component instances, it is also necessary to build different user interfaces for varying system configurations and user requirements. In a typical Web design environment, this would correspond to the design of individual pages and their sequencing or workflow management, with the additional complication that the pages are dynamically-generated by the back-end of a digital library system.

Figure 1 illustrates one view of the prototype system that was developed to design user interfaces for flexible digital libraries. In this prototype, the designer can lay out page elements as well as specify which services are to be incorporated and how the workflow among the pages of the interface will be effected, all through a Web-based interface. Formative evaluations were conducted through a series of participatory design sessions with stakeholders from different communities (e.g., digital library students, librarians).

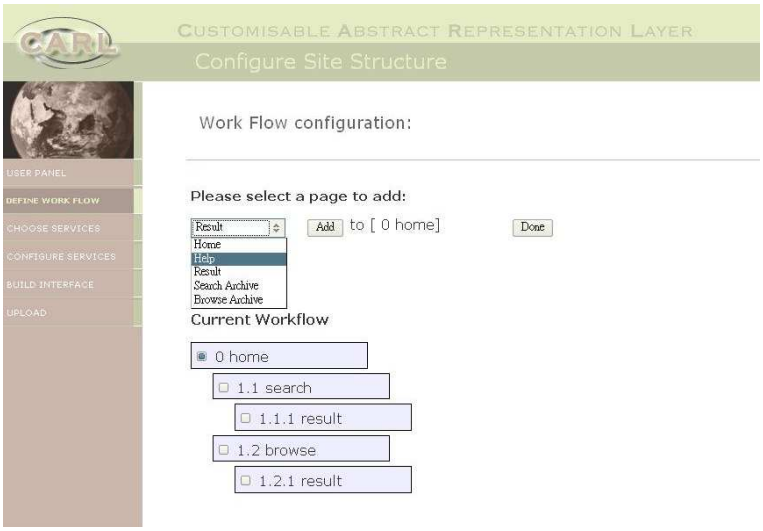

Fig. 1. User interface workflow editing

2.3 Flexible Component Packaging

As a final step in the process of making components appear as a cohesive whole, it should be possible to package a set of components, along with a description of their interconnections and a specification of the user interface(s) and workflows, into a single redistributable package.

A prototype packager and installer were developed to bundle a suite of components into a package for subsequent installation at a remote location. The packaging process allows a system designer to load a specification file, as output by the BLOX system, and then enter parameters and default values particular to the installation process.

Formal pilot studies have been conducted on the packaging and installing system and the feedback indicates that the system is preferable to one where individual components are installed and configured in isolation. Minor improvements have been made to the tools and further evaluation is planned for the near future. Like the first experiment, this study reinforces the need for components/instances to have a well-defined and standardised machine interface for configuration from an external source.

3 Conclusions

It is now widely accepted in the DL and Web Services communities that systems should be built as collections of loosely-connected communicating components. Much effort has already been expended on demonstration projects where components are used in innovative ways to build systems with different base requirements. It is time to move on to a higher level of design.

This paper reports on various studies that have built on earlier work in component technologies for digital libraries. These studies have all demonstrated the utility of and need for high level tools for the construction of digital libraries. In addition, they have uncovered the need for standard machine interfaces for the configuration and maintenance of components/instances.

In general, these experiments support the basic notion that components are an enabling technology to expand the boundaries of what is possible with information management and digital library systems.

4 Future Work

A study on scalability based on component farms is in its design phase and will test whether or not components provide an effective choice in granularity. There are still many unanswered questions and it is anticipated that much research will need to be done on how the component interfaces/protocols need to evolve and how services must be cast to get maximal benefit from cluster computing.

Existing production DL projects are, at the same time, gradually adopting component technologies and service-oriented architectures. The next version of Greenstone (v3) is being designed and developed according to a service-oriented

architecture for increased extensibility [1]. Similarly, DSpace is considering a far more modularised approach for its next generation [9].

Eventually, it is hoped that the higher level experiments with components discussed in this paper will contribute to an understanding of the pertinent issues in developing component frameworks so that production frameworks, such as the ones mentioned above, will be more robust and support a broad range of possible use cases.

5 Acknowledgements

This project was made possible by funding from UCT, NRF (Grant number: 2054030), NRF-THRIP, Telkom and Siemens.

References

1. Bainbridge, David, Katherine J. Don, George R. Buchanan, Ian H. Witten, Steve Jones, Matt Jones and Malcolm I. Barr (2004), "Dynamic Digital Library Construction and Configuration", in Heery, R., and L. Lyon (eds), Research and Advanced Technology for Digital Libraries: 8th European Conference (ECDL2004), 12-17 September, Bath, UK, LNCS 3232, Springer.
2. Castelli, Donatella, and Pasquale Pagano (2002), "OpenDLib: A Digital Library Service System", in Research and Advanced Technology for Digital Libraries, Proceedings of the 6th European Conference, ECDL 2002, Rome, Italy, September 2002, pp. 292-308.
3. DELOS (2001) Digital Libraries: Future Directions for a European Research Programme, San Cassiano, Alta Badia, Italy, 13-15 June 2001. Available http://delos-noe.iei.pi.cnr.it/activities/researchforum/Brainstorming/brainstorming-report.pdf
4. Eyambe, Linda K., and Hussein Suleman (2004), A Digital Library Component Assembly Environment. Proceedings of SAICSIT 2004, Stellenbosch, South Africa, pp.15-22.
5. Gladney, H., Z. Ahmed, R. Ashany, N. J. Belkin, E. A. Fox and M. Zemankova (1994), "Digital Library: Gross Structure and Requirements", Workshop on On-line Access to Digital Libraries, June 1994.
6. Kelapure, Rohit, Marcos André Gonçalves, Edward A. Fox (2003), Scenario-Based "Generation of Digital Library Services", in Proceedings of 7th European Conference on Digital Libraries (ECDL 2003), 17-22 August, Trondheim, Norway, Springer-Verlag GmbH, Lecture Notes in Computer Science, vol. 2769, pp. 263-275.
7. Suleman, Hussein, and Edward A. Fox (2001), "A Framework for Building Open Digital Libraries", in D-Lib Magazine, Vol. 7, No. 12, December 2001. Available http://www.dlib.org/dlib/december01/suleman/12suleman.html
8. Suleman, H. (2002), Open Digital Libraries, Ph.D. dissertation, Virginia Tech. Available http://scholar.lib.vt.edu/theses/available/etd-11222002-155624/
9. Tansley, Rob (2004), DSpace 2.0 Design Proposal, presented at DSpace User Group Meeting, 10-11 March, Cambridge, USA. Available http://wiki.dspace.org/DspaceTwo