Customising Interfaces to Service-Oriented Digital Library Systems

Hussein Suleman, Kevin Feng, Gary Marsden

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

Abstract. Digital library systems that once were mostly monolithic in construction are slowly making the transition to component-based models. However, it is not clear how best to design or construct the user interfaces to such systems - one alternative would be to create associated interface elements while another would be to create a separable interface layer. This paper discusses an attempt to do the latter by using current browser-based tools - recently named Ajax - in order to visually design the layouts, workflows and service connections of a user experience layer. Expert evaluators provided feedback during this process and the eventual level of functionality and usability of the proof-of-concept system demonstrate the inherent possibilities and relevance of the emerging Ajax technologies for not only the rendering or execution but also the design of browser-based Web applicatons, and digital library systems in particular.

1   Introduction

Current digital library (DL) systems such as DSpace, EPrints and Greenstone all require some - often non-trivial - customisation in order to fit in with the hosting organisation's Web presence. Simple changes such as HTML page titles are usually effected by the setting of parameters or variables. More elaborate changes such as integration with a university portal may require substantial (re-)programming.

In addition, users of standard toolkits may want to use a different set of services than those provided by default. This is especially relevant where a service-oriented architecture has been adopted and services can be readily added, removed or customised.

This high degree of flexibility has to be reflected in the user interface and most current DL systems do not cater for this. In contrast, the Web community has recently begun to create more flexible user interfaces using Asynchronous Javascript and XML (Ajax) [4]. Ajax is an approach to developing interactive browser-based user interfaces using a combination of Javascript and in-browser XML tools. Ajax applications are essentially Javascript applications associated with Web pages, with the added ability that they are able to send HTTP requests to Web servers and process the output in the Javascript code. Assuming

that the response is in XML, Javascript/Ajax provides the ability to access and manipulate the DOM tree of this XML response and/or the current document in the browser window, or perform transformations on any XML fragments using XSLT. Thus Ajax can be used to provide an interactive user interface within the otherwise static Web browser.

This paper reports on an attempt to use the Ajax approach to design a customisable user experience layer that caters for both interface and service flexibility. While Ajax has typically been used for rendering of interactive user interfaces, this work has attempted to use Ajax primarily as the basis of a design tool for user interfaces.

2   Service Oriented Digital Library Systems

In designing a user experience layer, it is necessary to connect interface elements to back-end services. For this, a Service Oriented Architecture (SOA) was used, as this is arguably ideal for loosely-coupled systems that need to interconnect with other systems. Greenstone 3 [1] uses such an architecture and DSpace [7] is possibly also going to adopt this approach.

User interfaces to SOA-based systems need to submit requests (typically in XML over the Web) and parse and reformat or transform responses (typically also in XML) in order to generate portions of the Web interface. This communication is easily accomplished using Ajax.

For this work, it was necessary to use a foundation set of services provided by an existing framework. The ODL [5] tools were used because of availability at the time, but the system could just as easily be layered over Greenstone 3. The ODL toolkit is a suite of components for providing DL services such as search and browse, each of which has a simple machine interface accessible over the Web [6] [2].

3   Design of the Designing System

The user experience layer was decomposed into the following three elements, with associated sections in the Ajax-based design tool:

- Services, for specifying and configuring a list of services, each of which connects to a service endpoint of a corresponding back-end service component;
- Flows, for specifying a list of pages and assigning a flow structure for inter-page navigation; and
- Pages, for designing each page using a WYSIWYG (visual) editor.

Figure 1 shows a screen snapshot of the visual page editor. Each page is designed as a series of elements that can be selected from a toolbar and dragged within a canvas. The toolbar contains both static (e.g., text) and dynamic (e.g., forms to invoke services) elements.

The flow structure editor then ties these pages together and provides a basic system of navigation. For example, the page where a user enters search queries

Fig. 1. User interface page layout editing

could lead to a page were search query results can be displayed. This structure is presented as a sitemap, using an approximation to a tree representation.

The services are connected into the individual pages when each page is being designed. Each service must, however, be configured in terms of the service endpoint and parameters necessary for the communication with this endpoint.

Finally, after the user experence layer has been designed, the designer can switch to the "playback" mode where the system is executed or rendered in production mode.

4 Evaluation and Analysis

The system was initially specified in a participatory design session with 3 post-graduate computer scientists, all with experience in digital library systems.

During its development, the system was subject to a two phase expert evaluation by a usability specialist and a digital library specialist. In the first phase various usability and functionality problems were identified and subsequently addressed [3]. During the second phase, it was felt that the functionality problems were largely resolved but the usability of the system could be further improved. However, the system was deemed to contain all the functionality required to customise and/or design a suitable user experience layer for a service oriented DL system.

This project has also led to some important observations about the Ajax technology:

– Javascript libraries are necessary to perform some functions that are sometimes taken for granted in traditional GUIs, such as drag-and-drop. While these libraries are readily available, they are not always easy to integrate at present.

– Ajax features are currently not available on all Web browsers, and where they are available they are not always consistent.
  – While a lot of user interaction is possible, the core HTML data format places restrictions on how data may be rendered e.g., drawing lines or graphs would require a lot of effort.

5 Conclusions and Future Work

This project has demonstrated that it is possible to create an interface for designing user interfaces to service-oriented digital libraries, using the Ajax approach. With some effort, the Ajax technologies have proven capable for the task but there is still room for improvement and greater standardisation. The ability to customise user interfaces using the interface itself shows promise for DL systems - users should be able to customise their interaction to some degree while designers should also not have to use lower-level tools for customisation or configuration.

6 Acknowledgements

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. Eyambe, Linda K., and Hussein Suleman (2004), "A Digital Library Component Assembly Environment", in Proceedings of SAICSIT 2004, Stellenbosch, South Africa, pp.15-22.
3. Feng, Fu-Yao Kevin (2006) Customisable Abstract Representation Layer for Digital Libraries, MSc Dissertation, Department of Computer Science, University of Cape Town.
4. Garrett. J. J. (2005), Ajax: A new approach to web applications, February 2005. Available http://www.adaptivepath.com/publications/essays/archives /000385.php.
5. 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
6. Suleman, H., and E. A. Fox (2002), "Designing Protocols in Support of Digital Library Componentization", 6th European Conference on Research and Advanced Technology for Digital Libraries (ECDL2002), Rome, Italy, 16-18 September 2002.
7. 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