in-Browser Digital Library Services

Hussein Suleman

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

Abstract. Service models for digital libraries have looked into how services may be decomposed into modules and components for greater flexiblity. These models are, however, mostly aimed at server-side applications. With the emergence of Ajax and similar techniques for processing XML documents within a Web browser, it has now become feasible for a browser to perform far more of the computational tasks traditionally encompassed in server-side DL services. Among other advantages, moving computation to the client can result in improved performance and scalability. As a new twist on service oriented computing, it is argued in this paper that digital library services can be provided partially or wholly through applications that execute client-side. Two case studies are provided to illustrate that such in-browser services are feasible and in fact more powerful and flexible than the traditional server-side service model.

1 Introduction and Motivation

A recent development in Web technology is the widespread acceptance of richer user interfaces based on in-browser processing of XML within embedded Javascript on webpages. This technology, dubbed Ajax, has within the last 2 years been adopted widely by digital library systems (DLSes) to enhance user interactivity by providing users with an experience that closely mirrors that of desktop applications.

Ajax, often defined as Asynchronous Javascript and XML [2], is based on various technologies built into modern browsers, including Javascript, the XML/XHTML DOM parser and mechanisms to load XML documents asynchronously. Ajax applications typically use Javascript to load remote XML documents asynchronously, then update the user's view of a webpage without contacting the server for a page reload.

In digital libraries, there have been some attempts to use Ajax to support more dynamic user interaction. Licsár et al. [3] used Ajax in their construction of a gesture-recognising information retrieval system for music. Their system communicated with a server and a video capture subsystem from within the Web-based user interface. Feng [1] showed that it was feasible not just to create DLS interfaces using Ajax but also to design the workflows and interfaces themselves in a browser, using Ajax techniques.

It can be argued, however, that Ajax techniques need not only be used for the uppermost veneer of DLSes (the user interfaces) and that it can be used also for service provision on the client's machine. This may have many advantages beyond the enhancement of user interaction. Firstly, the speed of interaction can be increased and the use of bandwidth decreased. More importantly, computation can be shifted to the client's machine, thereby freeing up the server and implicitly building greater scalability into the whole system. Figure 1 illustrates this shift in computation. In the traditional approach, most of the computation is concentrated on the server while in an in-browser service model, much of this computation happens on the client machine. In the latter case, the user interface is generated partially by server data and partially by the in-browser applications.

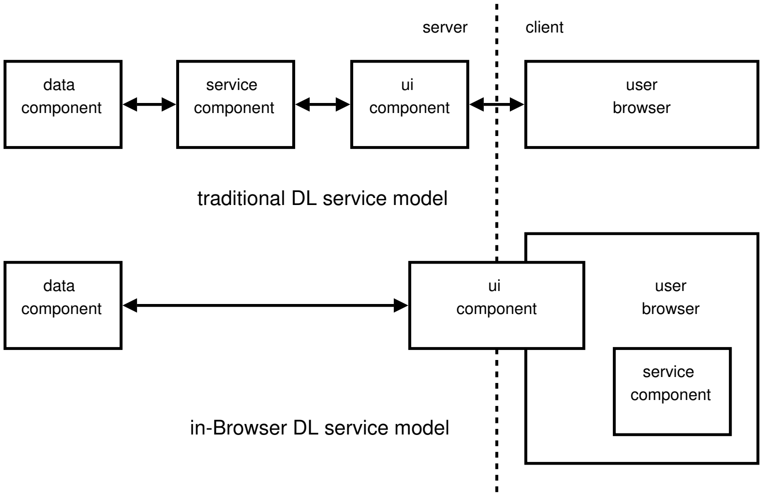

Fig. 1. Traditional and in-browser data/service provider models

Two case studies are discussed in this paper, illustrating how the notion of in-browser services can be effected using Ajax, as an alternative to server-side services.

2 Case Studies

2.1 RSS Validator

Really Simple Syndication (RSS) is a specification aimed at content syndication, where a data provider advertises a URL at which service providers may obtain a list of current or new content in a simple XML representation [6].

One of the biggest problems with RSS is a proliferation of specifications and some ambiguity about their interpretation, necessitating the use of external validators. Existing RSS validators are mostly server-side applications (e.g., http://feedvalidator.org/). An Ajax application was created to demonstrate similar functionality but as an in-browser service.

The RSS validator uses Ajax techniques to send a request for an RSS feed to a Web server. The XML response is then parsed and tested using a set of XML-DOM manipulations, with a report generated dynamically to the browser during the testing.

Security restrictions placed on Ajax allow only connections to the source (server or local disk) of the Web page. As a result, no Ajax-like methods will allow a user to load an XML document off a third-party server. This is not a problem in most digital library applications and third party validation was achieved in this case by using a simple HTTP proxy. If such a validation tool is packaged with DLS components, no proxy should be needed.

The validation tool was verified with 8 independently-coded local implementations from a simulated database of thesis metadata and 2 remote implementations (moodle, RSS Board Example) of the RSS 2.0 specification. Most implementations raised a few understandable warnings, such as date fields with incorrect formatting and inconsistent title information in different parts of the response. The level of error detection was deemed comparable to the existing online validation tools, but with all computation occurring in-browser.

2.2 in-Browser Search Engine

The Lucy Lloyd Archive Resource and Exhibition Centre recently commissioned the creation of a portable digital library system for the Bleek and Lloyd Collection [4]. This is a collection of notebooks and drawings that document the history and culture of some Bushman groups in Southern Africa. The data was collected primarily by Wilhelm Bleek and Lucy Lloyd in the late 1800s and serves as one of the few written records of this culture, which is widely recognised as among the oldest on our planet. This act of digital preservation has become far more urgent in light of the rapid assimilation of these ethnic groups into contemporary society [5].

The digital collection of notebook and drawing images is meant to be accesible online, offline or distributed on DVD-ROM, and should be usable irrespective of operating system or hardware architecture. This wide range of requirements is important to support the greatest possible audience of researchers wishing to use this data, even in remote parts of Africa. In this instance, Ajax was used to create a query service that is completely in-browser.

All data was first indexed by a standalone application and the inverted files and mapping of document identifiers to names (document index) were stored in static XML documents. Each inverted file includes part of the document index relevant to its documents. Thus, a single word query executes very efficiently as the document index is be optimal for it. Multi-word queries with overlap in document lists incur some penalty - it is known that most users prefer shorter queries so the problem ought to be minimal. This slight modification of a typical IR system increases the space required for the indices but decreases the time it takes to read in the document index for client-side applications.

3 Conclusions and Future Work

This paper has discussed an alternative approach to digital library services: that of executing service components on the client. By moving computation off the server, a number of benefits can be realised, including greater scalability and the possibility of self-contained in-browser digital library services. This also has implications for long-term preservation of services independently of the continued existence of service providers.

This refactoring of services has been demonstrated with 2 case studies using Ajax technology that is available in current browsers to convert typical server-side services into in-browser services. The case studies demonstrate that the idea of in-browser services is feasible in some cases and shows promise for a new twist on the paradigm of service-oriented digital libraries.

Ajax and similar client-side technology (such as XUL, Flash and Java) may offer particular benefits to DLSes and arguably should be integrated into current and future systems and design tools.

4 Acknowledgements

This project was made possible by funding from University of Cape Town, NRF (Grant number: GUN2073203) and the Lucy Lloyd Archive Resource and Exhibition Centre.

References

1. Feng, Fu-Yao Kevin (2006), Customisable Abstract Representation Layer for Digital Libraries, MSc Dissertation, Department of Computer Science, University of Cape Town.
2. Garrett, Jesse James (2005), Ajax: A New Approach to Web Applications, Adaptive Path, 18 February 2005. Available http://www.adaptivepath.com/publications/essays/archives/000385.php
3. Licsár Attila, Tamás Szirányi, László Kovács and Balázs Pataki (2006), Tillarom: an AJAX Based Folk Song Search and Retrieval System with Gesture Interface Based on Kodály Hand Signs, Proceedings of the 1st ACM International Workshop on Human-centered Multimedia (HCM '06), October 2006, ACM Press. Available http://doi.acm.org/10.1145/1178745.1178760
4. Lucy Lloyd Archive and Resource and Exhibition Centre (2007), Lloyd Bleek Collection, University of Cape Town. Available http://www.lloydbleekcollection.uct.ac.za/index.jsp
5. University of Cape Town (2003), Jewel in UCT's crown to be digitised for world's scholars, Monday Paper, 31 March 2003. Available http://www.uct.ac.za/print/newsroom/mondaypaper/?paper=114
6. Winer, Dave (2002), RSS 2.0 Specification, Berkman Centre for Internet and Society. Available http://blogs.law.harvard.edu/tech/rss