Report on Embedding and Reusing PerX in a VLE

By Santy Chumbe

10-May-2007

1. Introduction

This report presents the reusable middleware we have used to embed PerX functionality into the University VLE, VISION, a commercial VLE *Blackboard* system. We have done our best to use service-oriented architectures (SOA) as possible. We argue that by using open-source and open-standards approaches rather than software and practices developed specifically for a particular VLE product, it is possible to obtain open reusable middleware that can simplify the DL-VLE integration and bridge the functionality of both environments. We hope that our methodology can provide a common foundation on which a variety of institutions may build their own customized middleware to integrate scholarly objects in VLEs.

The work has been carried out within collaboration between the Heriot-Watt University Library and the Institute for Computer Based Learning (ICBL), with the aim to produce a *Blackboard* VLE - e-Library integration pilot using PerX software.

2. Methodology used for embedding the PerX toolkit into the VLE

The PerX toolkit has been already developed with an XML-based technology for system integration and a SOA approach, something, we need to recognize, it was difficult to always follow, as the toolkit is still a work in progress software. However, it is in some way a reusable library of open source software applications integrated by a SOA model and that we have combined via XML. Figure 1 represents the PerX toolkit architecture. Its main component is the *PerX Toolkit Engine*, which communicates with the rest of the software components via *wrappers*. The *wrappers* use XML-messaging for handling requests from/to the reusable APIs, which in turn deal with the database sources. The toolkit allows remote and local heterogeneous database sources to be cross-searched from one access point. It uses OAI-PMH for metadata exchange and SRU and Z39.50 for remote searching.

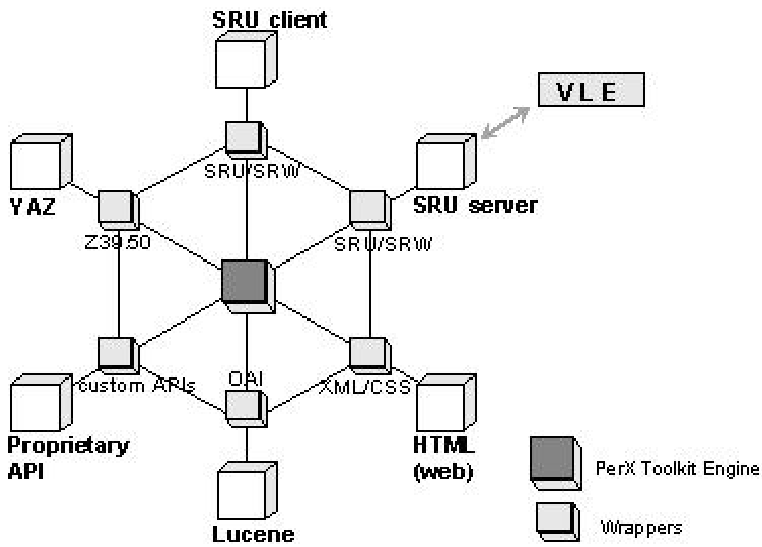

Figure 1 PerX Toolkit Architecture

The core software component of this pilot is the PerX toolkit described in the previous section. However the key player, or broker, of the integration itself is the reusable web-services based and SOA compliant middleware used to embed the toolkit functionality within the VLE. An important condition for the pilot was that the middleware should know nothing about the hosting VLE environment and thus can potentially be reused within any VLE framework. Its only function was to provide a "live bridge" between the toolkit functionality and the VLE system.

Our integration approach is based on a three-tier design pattern using an XML-based middleware API that sits between the *Blackboard* VLE (front end) and the PerX toolkit (back end) systems, and we do use the open standard SRU/SRW protocol for interoperability and XML message exchanging between the systems. The use of proven open standards effectively turns our XML-based middleware into a reusable *wrapper* or message broker. This approach would allow organizations to access virtually any SRU/SRW compliant system from within any SRU/SRW compliant DL system through a scalable service-oriented architecture. While simple, the middleware constitutes the basic infrastructure behind the DL-VLE integration, becoming a versatile alternative for integration. The basic deployment architecture of the proposed approach is shown in Figure 2.

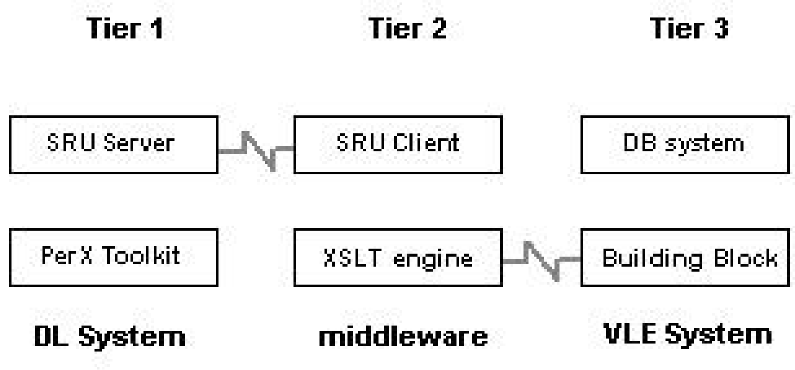

Figure 2 DL-VLE integration architecture using a three-tier design model

The approach described above offers a potentially rich system-level DL-VLE integration because it uses a standard specification for interoperability such as the Search/Retrieve via URL (SRU) protocol encapsulated in a reusable middleware. The SRU/SRW protocol is simple to implement because it is a standard REST-ful specification for providing Web Services functionality without the complexity of tightly coupled designs as found in remote procedure calls such as SOAP [1]. A REST-based protocol uses the HTTP mechanism to implement a client/server model using TCP/IP sockets [2]. The encapsulated middleware (the HTTP client) opens a connection to the PerX toolkit's SRU server (the HTTP server) and sends a request message consisting of a search query using the HTTP GET method. The HTTP server then returns a response XML message with the search results using the POST method and then closes the connection. The middleware then reformats the XML message and puts the search results into the VLE database system, so they can be shareable in the VLE modules. The middleware is a kind of proxy or intermediary software that handles requests on behalf of the systems that it is bridging.

In order that the middleware be recognized by the VLE as one of its components, it needs to be encapsulated in a *building block* of the *Blackboard Learning System*. This is accomplished by issuing an XML configuration file (*manifest*) to identify the middleware as a "bridge type" *Blackboard* "*building block*," and by including *Blackboard* proprietary Java class Tag libraries to abstract user interface components [3]. The middleware implementation has followed as strictly as possible the current Java Servlet specifications for Web applications [4]. The *Blackboard* system includes a portal running on a *Tomcat servlet* [5] and in fact its "*building blocks*" are just local *portlets* that can be handled as web applications individually deployed on the local *servlet*. These *building blocks* do not adhere to the web services specifications for remote *portlets* (known as Web Services for Remote Portlets WSRP specification [6]), so they are not shareable from other portals or remote systems. However, this is not an issue for our implementation as we supply the share-ability via the REST-ful model described above. At its very core, the middleware is an SRU client that provides standards-based technology to achieve integrated behavior and performance at the system-level across diverse environments such as the federated search toolkit and the VLE system.

To the best of our knowledge the approach of combining SOA with SRU to embed DL functionality within a VLE for cross-searching remote databases and local repositories has not yet been reported fully, and practical implementations have not received much attention. An attempt to integrate VLEs and digital repository systems using the SRU protocol in the open source d+ toolkit has been reported by Low B. [7]. However, some issues with the software were uncovered when using d+ for interoperability with VLEs. Despite claims of adherence to the current service-oriented trend, it was found that, apart from the SRU functionality provided by licensed OCLC software, deployment and use of the d+ toolkit required hardcode configuration of the software components as well as of the digital repositories. Also, at the time of testing, d+ could only query one database at the time (a sequential searching approach in contrast to the desirable "simultaneous" cross-searching approach.) Performance issues were also noted. It seemed that the ability of d+ for searching Z39.50 targets was bound to the limitations of the JAFER toolkit [8], which is still not fully available for production. Other open source alternatives considered before PerX and found unsuitable for the work discussed here, were the MDC toolkit, MyLibrary and the software suite Greenstone. In the UK, the JISC – DiVLE research strand involved a number of projects looking at how library resources can be integrated into VLEs using open standards. Thus, for example, between 2002 and 2004, the OLIVE project has been focused on how the OpenURL standard can be used to link Reading lists and Learning objects from the VLE. It also explored the use of Web Services (SOAP.) However, little practical achievement was reported [9], and unfortunately most of the plug-in software developed by the project was dependent on the commercial platforms used for integration (*MetaLib*, *Blackboard*, *Aleph*, *Discover*, *SFX*, etc.) For example, the method for implementing OpenURL is tightly coupled to the search form in the *Building Block* and cannot be reused for other applications. Also, the approach of the OLIVE project of loading functionality on the *Building Block* for metadata management raised many interoperability issues, as the *Blackboard* metadata functionality proved to be unusable and inaccessible to other areas within *Blackboard*. In Australia, Richardson J. [10] also reported on a project at Griffith University to integrate library resources into the *Blackboard* system. She recognized the power of commercial products in this arena, such as Sentient *Discover*, which supports OpenURL and Z39.50, but also highlights the “cognitive disconnect” faced by users of *Blackboard* when are taken away to the *Discover* user-interface environment from the *Blackboard* user-interface environment.

On the other hand, SOA approaches in e-learning are being promoted as suitable alternatives by important organizations such as JISC (the United Kingdom’s Joint Information Systems Committee), DEST (the Australian Department of Education Science and Training), ADL (the US Advanced Distributed Learning Initiatives), IMS (the Innovation Adoption Learning global learning consortium), NSDL (the US National Science Digital Library) and IC (Industry Canada). Need for stable and coherent technical frameworks or infrastructures where e-learning services can inter-operate harmoniously have been highlighted [11,12,13,14]. Our work is firmly in harmony with the above approach and recommendations, and it would be part of any standard e-learning framework where its functional components expose service behavior via loosely coupled interfaces. In this context, we follow with interest the work being carried out by related projects, such as the open source digital library architecture Fedora [15] and the NSDL Data Repository Architecture [16], as well as any research outcome from the JISC e-Framework for Education and Research Programme [17].

2. Study of the impact of DL-VLE integration on library users

The prototype working system demonstrating the VLE-Library system interoperability is being tested at the Heriot Watt University. Testing is being carried out with a group of academic library users, and feedback is being gathered using a short questionnaire and informal interview. We hope to use the outcomes of the testing to asses the impact of the VLE-library integration on academic and library users as well as a basis for gathering suggestions and recommendations for future developments to benefit institutional planning for library and institutional VLE integration. The reuse and sharing of DL content among the different VLE components is being explored with particular interest. We have had high interest in finding out how users rate the usefulness of cross searching from within the VLE and the convenience of onward use of search results in other VLE functions e.g. exporting, saving, emailing and posting them to discussion boards. The prototype system, named as *PerX Building Block*, provides distributed searching of a sample of subscribed e-journals, the local library catalogue (OPAC) and the Google search engine.

Regarding possible performance issues of the proposed DL-VLE integration, we have noticed that SRU/SRW is not necessary relatively slow. We were expecting that the SOA-based prototype be significantly slower than fast, general purpose search engines because it uses XML-based messaging services, which typically consume more computing resources. However, after assessing the performance of the search services when searching various heterogeneous scholarly objects, users noticed that speed and performance were not issues in the prototype.

Finally, some recommendations for increasing the usability and the effectiveness of the prototype have been identified. In addition to more sophisticated retrieval and searching algorithms (e.g. full common Boolean

support across heterogeneous databases), there are key operational enhancements that have been acknowledged as desirables. Enhancements include:
* Combining search results from multiple databases, which involves unified ranking.
* Comparing and consolidating search results (simplest case: removing duplicate search results; more complex case: fussy techniques for combining several databases´ results).
* Discovering inconsistencies and removing them in the search results (for example search results that seems to be different but in fact point to same resources).

6. Conclusions

By using open-source and open-standards approaches rather than products and practices developed specifically for an individual VLE product, we have obtained a reusable middleware that can provide a common foundation on which a variety of institutions may build their own customized middleware to integrate their scholarly objects in VLEs. Our study hopes to demonstrate that the use of service-oriented architectures (SOA) and REST-ful based (SRU) open source middleware is a cost effective, simple and open alternative for embedding digital library services within learning and teaching frameworks.

Although SOA middleware reduces the need for system development and also management and maintenance burdens, the performance of SOA-based search services need to be monitored for large production services, because XML-based messaging services typically consume more computing resources and are slower than fast general purpose search engines. Early testes suggested that users have not found performance issues using the DL-VLE integration prototype system.

Acknowledgements

On behalf of the PerX Project we acknowledge the efforts and help of the Heriot Watt University Library staff as well as give special recognition to Dr. David A. Cole for his fundamental role in the installation of the middleware on the VLE described here.

Notes and References

[1] zur Muehlen M., Nickerson, J,; Swenson, K. *Developing Web Services Choreography Standards – The Case of REST vs. SOAP*. Decision Support Systems 37. Elsevier. 2004.
[2] Fielding R. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral Dissertation. University of California, Irvine, CA (180p.) 2000
[3] Blackboard Building Blocks. Developer Guide. Blackboard Learning System Release 7.1 2006
[4] JSR 154: JavaTM Servlet 2.4 Specification: http://jcp.org/en/jsr/detail?id=154
[5] Apache Tomcat Servlet: http://tomcat.apache.org/tomcat-6.0-doc/index.html
[6] Polgar J., Bram R., Polgar A. *Building and Managing Enterprise-wide Portals*. Idea Group Inc (IGI) 2006
[7] Low B., MacColl J. *Searching Heterogeneous e-Learning Resources*. Paper presented at the DELOS Workshop 2005, Digital Repositories: Interoperability and Common Services, May 2005
[8] JAFER Toolkit Project: http://www.jafer.org
[9] OLIVE Project Final Report (2004) [Available at http://www.jisc.ac.uk/uploaded_documents/OLIVE_Project_Report.pdf]
[10] Richardson J. *Building Bridges between Learning Management Systems and Library Content Systems*. Presented at the 11th Australian World Wide Web (AusWeb05) Conference, Gold Coast. 2005 [Available at http://ausweb.scu.edu.au/aw05/papers/refereed/richardson/index.html]
[11] S. Wilson, K. Blinco, D. Rehak. *Service-Oriented Frameworks: Modelling the Infrastructure for the Next Generation of e-Learning Systems*. Presented at Alt-I-Lab Conference, 2004
[12] Powell, A. *A 'service oriented' view of the JISC Information Environment*. UKOLN Report. Nov. 2005 [Available at http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/soa/jisc-ie-soa.pdf]
[13] Payette S. *Choosing Technology that can Evolve With User Needs. A service-oriented approach to e-research, e-scholarship, and advanced scholarly publication*. VALA 2006 Conference. Melbourne, Australia. February 2006 [Available at http://www.valaconf.org.au/vala2006/papers2006/97_Payette_Final.pdf]
[14] Hunter J. *Scientific Models – A user-oriented approach to integrating scientific data and digital libraries*. VALA 2006, Melbourne. February 2006 [Available at http://www.valaconf.org.au/vala2006/papers2006/55_Hunter_Final.pdf]

[15] Johnston L. *Development and Assessment of a Public Discovery and Delivery Interface for a Fedora Repository*. In D-Lib Magazine, Vol. 11, Number 10. 2005 [Available at http://www.dlib.org/dlib/october05/johnston/10johnston.html]

[16] Lagoze C, Kraffti D., Payettei S. and Jesurogaii S. *What Is a Digital Library Anymore, Anyway? Beyond Search and Access in the NSDL*. In D-Lib Magazine Vol. 11, Number 11. 2005 [Available at http://www.dlib.org/dlib/november05/lagoze/11lagoze.html]

[17] The JISC e-Framework for Education and Research: http://www.jisc.ac.uk/whatwedo/programmes/programme_eframework.aspx