The Potential for Web Services to Enhance Information Access to Legacy Data: An Exploratory Study and Application

Fatih Oguz
Research Associate - Texas Center for Digital Knowledge, University of North Texas, PO Box 311068 Denton, TX 76203. Email: foguz@lis.admin.unt.edu
William E. Moen
Associate Professor - Texas Center for Digital Knowledge, University of North Texas, PO Box 311068 Denton, TX 76203. Email: wemoen@unt.edu

This poster presents an overview of an exploratory research project to identify, describe, and investigate the applicability of a Web services (WS) approach to access legacy data. In the Z Texas Implementation Component of the Library of Texas (ZLOT) project, the ZLOT technical team implemented a multi-purpose Texas Library Directory Database (TLDD) that is used as a back-end database to support the Library of Texas (LOT) Resource Discovery Service (RDS). The researchers developed and implemented a prototype WS application to show how a legacy system can be accessed and its data can be searched and retrieved. This study focused on understanding how requests and responses between software applications are encoded in Extensible Markup Language (XML).

Introduction

RDS is implemented by using the PHP scripting language and XML. The TLDD is a robust MySQL relational database that includes vital information about Texas academic and public libraries (Lopatovska *et al.*, 2004). It would be useful if the TLDD can be made available to the Texas library community in a way that librarians can query the database and get results in structured XML documents for reuse in other applications. Currently this service is not available. A WS application appears to offer a potentially useful approach for such interaction with the TLDD.

WS are defined as software systems designed to support interoperable machine-to-machine interaction over a network by using XML for sending and receiving messages. The WS are self-contained applications that can be described, published, invoked, and located over the Internet (or any network). Once a Web service is deployed, other applications can discover and invoke the service. WS provide a programmable interface for other applications without requiring custom programming and proprietary solutions regardless of the operating systems and programming languages to share information as opposed to providing users with a graphical user interface (Boss, 2004). XML, Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL) are tools to create a WS.

The assumption for this project is that a WS application can offer a reliable, flexible, and standards-based solution for accessing the TLDD by the library community to search and retrieve structured and reusable data.

XML

The elegant simplicity and flexibility of the XML made it a definitive standard for data transmission and storage. XML is an open standard and can be accessed and processed by any tool capable of reading and writing American Standard Code for Information Interchange (ASCII) text. XML is a cross-platform, software, and hardware independent tool to transmit data in plain text format.
XML allows data and content to be packaged in a common format that is machine-readable and is able to be manipulated between similar or different application environments. XML lets applications present data, syntax, schema and semantics when sending data. Element and attribute names identify data in an XML document and data type are defined by using XML Schema Definition (XSD) or Document Type Definition (DTD). XSD lets services and clients running on various platforms semantically interoperate over a common document structure (Burner, 2003, "Web services gotchas", 2002). XSD is an XML based alternative to DTD to define the structure of an XML document.

SOAP

SOAP is an XML-based communication protocol that is vendor neutral, programming language independent, and transport and network protocol independent. SOAP is simple communication protocol that defines a message structure including optional Header element and mandatory Body element, wrapped up by an envelope (Burner, 2003). A well formed XML fragment enclosed in SOAP elements (e.g., Envelope, Body) constitutes a SOAP message. XML documents can be generated easily using any kind of programming language.
The primary use of SOAP is to make different programs running on different platforms able to communicate. SOAP is widely used because it supports interoperability among many different

environments and it can be used in combination with various protocols, for example, Hyper Text Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and Transmission Control Protocol (TCP).

WSDL

WSDL is an XML-based abstract description of a WS. WSDL provides a map or template showing how services should be described and used by clients (Vasudeman, 2001). WSDL plays a key role in ensuring technical level interoperability by describing how other application can consume the service. A client can locate and invoke a WS by using WSDL document. A WSDL document specifies data types and messages to be transmitted, supported operations, SOAP specific information, and location of the service (e.g., URL).

Texas Library Directory Web Service

Search/Retrieve Web service (SRW) is a WS standard to perform searches and other information retrieval operations on the Internet using SOAP. SRW has been built on 20 years of experience with the Z39.50 information retrieval protocol (Sanderson, 2004). SRW uses various schemas such as Dublin Core and it allows developers to define their own schemas as well. SRW uses a query language called Common Query Language (CQL) that offers simplicity and intuitiveness of Common Command Language (CCL) as well as power and expressiveness of Structured Query Language (SQL). CQL is a formal query language to express searches on Web indexes, bibliographic catalogs and museum collection information (Sanderson, 2004). The SRW 1.1 protocol has been adopted in the implementation of Texas Library Directory Web service (TLDD WS).

CQL provides users with great advantages over traditional query languages such as SQL. In this research project we mainly focused on the simplicity and functionality of the WS rather than its incorporation with other technologies like CQL. However, TLDD is a MySQL database and it can be queried using SQL (Lopatovska et al., 2004). Although a WS approach hide all the complexity behind this pilot application, researchers did not implement all the features of CQL standard. The pilot application incorporated an external WS called VB CQL Parser to parse CQL queries and convert into XML Query Language (XQuery) (Habing, 2004). Later XQuery has been translated to SQL statements for querying the MySQL database. This service is a Search and Retrieve URL (SRU) service, a companion to SRW, which passes parameters via URL instead of SOAP and the response returned in XML format. While transforming CQL statements to SQL researchers were not able to support some of the features provided by CQL (e.g., proximity operators, fuzzy search) because of the limitations in MySQL that does not support such kinds of operations. The pilot WS application supports simple term and simple Boolean searches.

How TLDD WS Work?

The WSDL file defines how other SOAP clients can communicate with the TLDD WS.

The TLDD WS is running on Linux and has been implemented using PHP scripting language and NuSOAP toolkit. NuSOAP is a set of PHP classes that let developers create and consume WS (Nichol, 2004). On the server side, NuSOAP uses WSDL calls included in the WS code to generate the WSDL document automatically. On the client side, regardless of the operating system (e.g., Windows, Linux) a client SOAP toolkit compiles this WSDL document to generate all the code a client needs to invoke the service and process the response.

The SOAP client sends a SOAP request to TLDD SOAP server then it passes the request to the server side application. A simple search and retrieve request is composed of three basic parameters of version, query, and number of records to be returned. The CQL query is transmitted to an SRU service for processing, and the returned XQuery is parsed and converted to SQL to query TLDD. Finally, results are passed back to the SOAP server and it sends the response back to the SOAP client as an XML document.

References

Boss, R. W. (2004). Web services. Retrieved November 1, 2005, from http://www.ala.org/ala/pla/plapubs/technotes/webservices.htm

Burner, M. (2003). The deliberate revolution: Transforming integration with xml web services. *ACM Queue, 1*(1).

Habing, T. G. (2004). Experimental oai registry at uiuc. Retrieved April 15, 2005, from http://gita.grainger.uiuc.edu/registry/sru/sru.asp

Nichol, S. (2004). Programming with nusoap using wsdl. Retrieved February 15, 2005, from http://www.scottnichol.com/nusoapprogwsdl.htm

Sanderson, R. (2004). Srw: Search/retrieve webservice version 1.1. Retrieved Februrary 2, 2005, from http://srw.cheshire3.org/SRW-1.1.pdf

Vasudeman, V. (2001). A web services primer. Retrieved February 10, 2004, from http://webservices.xml.com/pub/a/ws/2001/04/04/webservices/index.html?page=1

Web services gotchas. (2002). Retrieved March 5, 2004, from http://www-306.ibm.com/software/solutions/webservices/pdf/bloor_gotchas.pdf