# Texas Library Directory Web Services Application: The Potential for Web Services to Enhance Information Access to Legacy Data

By: Fatih Oguz and William E. Moen

Oguz, F., & Moen, W. E. (2006). Texas Library Directory Web Services Application: The Potential for Web Services to Enhance Information Access to Legacy Data. *Proceedings of the International Conference on Next Generation Web Services Practices (NWeSP'06).* Seoul, Korea.



## Abstract:
This paper presents an overview of an exploratory research project to identify, describe, and investigate the applicability of the 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 has implemented a multipurpose 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 XML.

## Article:
INTRODUCTION
Client/server networks, dominant in the 1980s and early 1990s, are evolving into a new kind of networked environment built around a set of open protocols and standards including HTTP, SOAP, XML, and UDDI. According to Coyle, the emerging Web Services (WS) technical infrastructure ensures that applications and services from different vendors will interoperate to support one or more business processes [1].

If data can be exposed in XML, it appears that WS provide an opportunity to expose legacy data to applications and services. In the context of this project, legacy data are defined those that need to be converted to a different format when another application or service will ingest, consume, or otherwise use the data [2].

For this project, a relational database containing directory information about Texas academic and public libraries was used as a target for the implementation of Search and Retrieve URL/Web Service (SRU/SRW) which are WS for search and retrieval based on Z39.50 semantics [3]. The target database, the Texas Library Directory Database
(TLDD), was developed as part of a larger application (see below), but it contains useful information about Texas libraries that could be reused and repurposed if the TLDD information could be provided in a standard structure using XML. The goal of this project was to investigate the extent to which the SRU/SRW WS could be used to search the relational database and return records formatted according to an XML schema to enable reuse of those data.

## BACKGROUND ON THE LEGACY DATA AND CURRENT PROJECT

The Texas State Library and Archives Commission (TSLAC) developed and implemented a metasearch application called the Library of Texas (LOT) in 2002-2003 [4]. The LOT comprises several basic components. One component is the Resource Discovery Service (RDS) that provides users a mechanism to search multiple online library catalogs, commercial databases, and other digital resources from a single search interface. Another component is the TLDD, which contains directory information about Texas libraries. Data from the TLDD is used by the RDS to dynamically customize the RDS search interface. The TLDD is a robust and complex MySQL database [5]. TSLAC contracted with Index Data <http://www.indextdata.com> to implement the RDS using open source tools and code. The RDS source code is available to TSLAC, which is making it available to Texas libraries that choose to use it as a single search interface to locally licensed resources and other distributed or local digital resources. However, since the TLDD was a key component for powering the RDS, local libraries wanting to use the RDS faced the challenge of how their locally installed RDS applications could access the TLDD in a standardized and interoperable manner.

Our project assumed that a WS application could offer a reliable, flexible, and standard-based solution for accessing the TLDD, not only by local RDS applications but also by the library community, to search and retrieve structured and reusable data. Specifically, if the TLDD data could be presented to other applications in an XML format, reuse and repurposing of the TLDD data could be enhanced. The SRU/SRW WS would provide the means for querying the TLDD, and a XML schema specific to the TLDD data would provide the standard structure for encoding the data to return to processing applications.

## SEARCH AND RETRIEVE URL/WEB SERVICE

Search and Retrieve URL/Web Service consists of two standards used to perform searches and other information retrieval transactions on remote information retrieval and database systems. The SRU standard issues the query parameters in a structured URL over HTTP. For the current project, we investigated the use of SRW, which uses SOAP and the XML Path Language (XPath) SRU/SRW has been built on 20 years of experience with the Z39.50 information retrieval protocol [6]. SRU/SRW uses various schemas such as Dublin Core and allows developers to define their own schemas for encoding data to return to an application in response to a search.

SRW enables developers to implement a standard-based search interface to information retrieval systems easier than with the more complex Z39.50. SRU/SRW uses a query language called Common Query Language (CQL). CQL is a formal query language used to express searches on Web indexes, bibliographic catalogs and museum collection information [5]. CQL offers the simplicity and intuitiveness of Common Command Language (CCL) as well as the power and expressiveness of Structured Query Language (SQL). Both SRW (Version 1.1) and CQL (Version 1.1) were released in 2004 [6].

We adopted the SRW 1.1 protocol in the implementation of the Texas Library Directory Web service (TLDD WS). The following sections describe the operations available via SRW, namely SearchRetrieve, Scan, and Explain. Each of these operations has corresponding XML schemas to encode the protocol requests and responses.

### *SearchRetrieve*

The SearchRetrieve operation consists of a request and response that enables users to issue a query to a remote database and retrieve records that match the search criteria. A SearchRetrieve Request can contain eleven defined parameters, two of which are mandatory. Similarly a SearchRetrieve Response may contain nine parameters, two of which are mandatory.

In our TLDD WS application, the SearchRetrieve Response includes the parameters mentioned below for testing purposes only (these parameters are not defined as part of the SRU/SRW standards):

- index: name of the index of the returned records
- responseTime: measures response time in seconds for the processed query

### *Scan*

The Scan operation enables users to browse terms in one or more indexes available from the information retrieval application. The user submits a term and the response can contain a list of terms, above or below the submitted term, in the index; however, a list of terms is not a mandatory parameter for the Scan Response. A Scan Request may contain six parameters, two of which are mandatory. A Scan Response may contain five parameters, one of which is mandatory.

### *Explain*

The Explain operation enables user to know what a SRW server/database supports in terms of the protocol and CQL. An Explain Request may contain four parameters, one of which is mandatory.

The Explain Response may contain five parameters, two of which are mandatory. The ‘record’ parameter must contain a ZeeRex record. A ZeeRex record is an XML document formed according to the ZeeRex DTD and describes server/database characteristics such as indexes available for searching, schemas supported, etc. Our pilot application adopted version 1.0 of the ZeeRex DTD and returns brief records [7].

## TLDD WS COMPONENTS AND OPERATIONS

The TLDD WS is running in a Linux environment and is implemented in PHP scripting language. The WSDL file, which defines how other SOAP clients can communicate with the TLDD WS, resides on the TLDD server. NuSOAP generates the WSDL. A SOAP client makes a request for the WSDL file, which is then returned by the server, Figure 1 illustrates how a WSDL file is created and consumed by other SOAP clients, how SOAP requests and responses communicated between a SOAP client and a server, and how SOAP requests are evaluated on the server side. On the client side, the operating environment could be Windows, for example, and client SOAP toolkit compiles the WSDL to generate all the code the client needs to invoke the service and process the response.

The SOAP client sends a SOAP request to the TLDD SOAP server. In that SOAP request is the SRW requests (e.g., SearchRetrieve, Scan, or Explain), which is passed from the SOAP server to the server-side application. In response, an XML document is returned according to the SRW schema specifications.

For a search and retrieve operation, the SRW SearchRetrieve Request embedded within a SOAP request is submitted by the client. The SOAP server evaluates the request, passes the query parameters to the server-side application. An external SRU service is incorporated to process CQL queries. CQL queries submitted by the client transformed into XQuery via an SRU transaction to support a search of the MySQL TLDD. The returned XQuery is parsed and converted to SQL statements to query the TLDD.

The search results from TLDD are then formatted into an XML document according to the TLDD schema for retrieved records, and are passed back to the SOAP server. It returns a SOAP response to the SOAP client which then processes the results.

With this implementation, we have made the TLDD available as a WS, and the data returned from the TLDD are in a standardized, structured XML format for use by other applications, such as displaying the data, transferring the data to another application, etc. The TLDD is no longer locked in a legacy system, but available for reuse, repurposing, etc.

## ISSUES OF CONCERN WITH WEB SERVICES

Although WS offer organizations and programmers great advantages over existing systems, several important issues have already been identified and addressed by the W3C and WS community [8].

### *Security*

Security is the most important shortcoming of the WS technology. Most of the current WS implementations rely on third-party software solutions to secure their WS, if necessary. Since this WS project was a pilot, we did not employ third-party software other than what was already in place on the server where WS files were hosted (e.g., firewall).

For document level security, the data released from the TLDD did not contain any security sensitive information that would require a security protocol (e.g., encryption). However, various XML protocols exist that could provide document level security, such as XML Encryption and XML Signature. Since researchers focused on the basic functionalities of the WS technology rather than using additional standards and software to implement a fully functional and secure WS, the pilot project was implemented without additional security features.

Moreover, additional security features, if necessary, could be added to the existing pilot implementation, which currently provides an underlying infrastructure for a fully functional TLDD WS. However, the NuSOAP 0.6.9 implementation does not conform to the WS-Security profile.

### *Messaging*

Some WS providers may require an access key and limited number of requests within a certain period of time (e.g., Google API, Amazon’s ECS) not only to authenticate users but to prevent their systems becoming overloaded due to possible denial of service attacks.

The SOAP 1.2 standard addresses the security issues in the SOAP message exchange protocol. The latest NuSOAP version was 0.6.9 at the time of this project and it only supports SOAP 1.1 specifications.

SRW (Version 1.1) used in this project does not address the above mentioned issues. Nevertheless, user authentication and access control could be monitored with an additional layer of WS. Users could access the TLDD WS after a verification process.

NuSOAP does not provide any asynchronous SOAP calling. Instead, when a SOAP call is placed, NuSOAP will not return anything until the call is responded to or time-out occurs.

### *Reliability*

Reliability in messaging can be ensured with the adoption of a different SOAP toolkit that supports the SOAP 1.2 standard. In addition, a logging system could be implemented as another layer to monitor the transactions.

### *Interoperability*

The messaging protocol is mainly responsible for interoperability. SOAP is responsible for both generating WSDL documents and consuming WSDL documents generated by others.

A WSDL file can be created by either a SOAP toolkit or manually by the developer. Although complying with WS-I profile does not guarantee interoperability of the WS with other applications, it can serve as a testing tool that ensures a certain level or degree of interoperability. However, a WS conforming to the WS-I profile may not support the clients built-in the .NET 1.1 framework. The current state of WS technology may provide interoperability to a certain degree with given standards. We tested the TLDD WS with various SOAP toolkits to ensure the interoperability. XML SPY 5 (Enterprise Edition), Generic SOAP Client [9], and Mindreef SOAPscope [10] SOAP clients successfully generated the client side codes using the WSDL file created by the NuSOAP and consumed the TLDD WS.

### *Performance*

In SRU, a request is encoded within the URL then transferred via HTTP GET method which does not employ SOAP.

In our pilot, we employed a SRU service to convert CQL queries to XQuery. Having an SRU service integrated into the system did not affect the overall performance of the application. However, adding various security features may have a negative effect on the performance.

## PROJECT EVALUATION

The purpose of this project was to investigate the applicability of WS technology in accessing legacy data. The application itself is not robust and provides limited functionality (e.g., CQL), however, this was not due to WS architecture but rather programming limitations. The application implementation process showed that developing such services does not require allocation of vast resources and proprietary toolkits. NuSOAP was selected not only because of its simplicity and flexibility but also because of its popularity among WS developers [11]. However, we were not able to discover sufficient documentation for the NuSOAP toolkit; instead

we found many sample scripts and a limited number of short articles. The NuSOAP toolkit offers more features than we used in the pilot implementation due to time and resources constraints available for the project. We focused only on the features needed to develop the application.

Even though the development cycle of TLDD WS was relatively short, we spent a great deal of time figuring out how to use PHP in WS development. Once we were comfortable with the technology, development was quick and easy.

Although the simplicity and flexibility of NuSOAP enabled us to implement the TLDD WS in a short period of time, other mature SOAP toolkits (e.g., Apache Axis) should also be considered in future projects. The current NuSOAP implementation may fall short in meeting required security and interoperability standards for mission critical WS applications especially for those made available to third parties. NuSOAP may be a good candidate for WS within an organization where organizational network security would already be ensured by additional software.

In terms of the SRW component of the TLDD, a few items need to be mentioned. CQL provides users with more advantages than traditional query languages such as SQL. In this project we focused primarily on the simplicity and functionality of the WS rather than its incorporation with other technologies like CQL. TLDD is a MySQL database and it can be queried using SQL. Although the WS approach hid all the complexity behind this pilot application, we did not implement all the features of CQL standard. The pilot application incorporated an external WS (i.e., VB CQL Parser) to parse CQL queries and convert into XQuery. This service is a Search and Retrieve via URL service. The returned XQuery is translated to SQL statements for querying the MySQL database. While transforming CQL to XQuery to SQL statements, we were not able to support some of the features provided by CQL (e.g., proximity operators, fuzzy search) because of the limitations in MySQL database, which does not support such operations. The pilot WS application supports simple term and simple Boolean searches.

## CONCLUSION

The results of this exploratory project demonstrated the applicability of WS protocols to operational systems in the library community. In this study, a pilot application of a search and retrieval WS against a legacy database (i.e., the TLDD) has been implemented in an open source environment (i.e., using Linux, MySQL, and an Apache web server). Further research should be carried out to investigate the scalability of the software and the levels of interoperability with SOAP implementations other than those tested in this project.

The project was undertaken in the context of the LOT's RDS and its reliance on the TLDD for data to power and customize the RDS. Local libraries may be more interested in implementing the LOT's RDS application since they could reuse TLDD data via the TLDD WS in their local RDS implementations.

The relevant literature suggests that WS technology is likely to be adopted by organizations of various sizes and is expected to play a key role in information systems, but it will not replace expensive, proprietary systems like Electronic Data Interchange (EDI) in the near future because of their large user base [12]. WS technology can be used as a complementary tool for current EDI implementations. According to a Yankee Group survey conducted in 2004, forty-eight

percent of the US enterprises have already deployed WS with another thirty-nine percent planning to do so even though most of these WS implementations are at the experimental level [13]. The user base of WS technology is getting larger day by day, and the library community should not hesitate to adopt this technology for the different services they offer (e.g., Inter-Library Loan). WS technology appears to be the best candidate for accessing the legacy data and for application integration due to its low learning curve, short development cycle, and quick return on investment. Thanks to Library of Congress' SRW initiative, libraries will be able to preserve their current technology investments (i.e., Z39.50) while adopting this new technology.

Finally, this study may provide new perspectives for future projects on RDS and TLDD where WS could be used as an underlying technology in ensuring interoperability.

## FUTURE STUDY

Since the purpose of the study was not to create full scale WS implementation, researchers focused on basic functionalities to develop the application. Adoption of CQL and complying with SRW 1.1 requirements would offer great opportunities for library community, since TLDD stores very valuable information for librarians, publishers and so on. Libraries and consortia should outreach their users to learn what kinds of services should be made available to the public and developers. Possible cost savings should be investigated in library information systems.

In addition, representation of powerful and complex CQL queries in SQL should be studied further.

## ACKNOWLEDGEMENT

The School of Library and Information Sciences at the University of North Texas funded this study through the Faculty/Ph.D. Student Grant Program in an effort to socialize doctoral students into the research process, as well as a means for fostering interdisciplinary activity with the Interdisciplinary Information Science Doctoral Program.

## REFERENCES

1. F. P. Coyle, “"XML, Web Services and the Changing Face of Distributed Computing,” ACM Ubiquity, vol. 3, 2003. [Online]. Available: http://www.webservices.org/index.php/article/articleview/75. [Accessed October 27, 2003].

2. P. Dyson, D. Chang, R. Godwin, and P. Weinberg. “Repurposing Legacy Data: Everybody's Business.” [Online]. Available: http://seminars.seyboldreports.com/1998_san_francisco/ETAPE_48.html. [Accessed January 7, 2006]

3. Library of Congress. SRU: Search and Retrieve via URL. 2006. [Online]. Available: http://www.loc.gov/standards/sru/ [Accessed August 29, 2006].

4. Library of Texas.” [Online]. Available: http://www.tsl.state.tx.us/lot/overviewlib.html. [Accessed February 18, 2005]

5. I.V. Lopatovska, F. Oguz, and W. E. Moen, “Design, Development, and Implementation of a Texas Library Directory Database: A Multipurpose Database for the Library of Texas,” presented at ASIST 2004 Annual Meeting; “Managing and Enhancing Information: Cultures and Conflicts”, Providence, RI, 2004.

6. R. Sanderson. “SRW: Search/Retrieve Webservice Version 1.1.” [Online]. Available: http://srw.cheshire3.org/SRW-1.1.pdf. [Accessed February 2, 2005]

7. “ZeeRex: The Explainable ``Explain" Service.” [Online]. Available: http://explain.z3950.org/. [Accessed February 2, 2005]

8. Bloor Research N.A., “Web Services Gotchas,” May 2002. [Online]. Available: http://www.306.ibm.com/software/solutions/webservices/pdf/bloor_gotchas.pdf. [Accessed: March 5, 2004]

9. “Generic SOAP Client.” [Online]. Available: http://soapclient.com/soaptest.html. [Accessed March 1, 2005]

10. “Mindreef: Comprehensive Web services diagnostics and testing.” [Online]. Available: http://www.mindreef.net/tide/scopeit/start.do. [Accessed: March 1, 2005]

11. S. De. “Sun Leads Even in the Web Services Tool Category,” February 2005. [Online]. Available: http://www.developer.com/services/article.php/3483101. [Accessed July 15, 2005]

12. J. L. Siegrist, “An Evaluation of Web Services with Emphasis on Small Hotels,” Bournemouth University, Dorset, United Kingdom, 2005.

13. D. Barlas, "Web Services Update," in Line56.com, 2004. [Online]. Available: http://www.line56.com/articles/default.asp?ArticleID=6028&TopicID=4. [Accessed January 1, 2005]