An Interoperable Portal for the Historic Environment

Francisco Pinto*<fqp1@ukc.ac.uk>, Nick Ryan <n.s.ryan@ukc.ac.uk>,
Tony Austin <afa2@york.ac.uk> and Julian Richards <jdr1@york.ac.uk>

Abstract

Research into the Historic Environment is particularly concerned with the search for information resources using spatial and chronological attributes. A recent pilot project between the Computing Laboratory of the University of Kent at Canterbury (UKC) and the Archaeology Data Service (ADS) at the University of York has successfully developed a Portal that provides Z39.50 enabled searching of a number of geographically remote data sources. The ADS provides access to some 400000 index records about the Historic Environment, nearly all of which are spatially and temporally referenced. Other partners in this project include the Portable Antiquities Scheme of British Museum (PAS), the Royal Commission on the Ancient and Historic Monuments of Scotland (RCAHMS) and the Scottish Cultural Resource Access Network (SCRAN) who, along with the ADS, will act as targets for a Historic Environment Portal.

The Portal allows the virtual searching of the holdings of the partner organisations as one. It has options to search on, in any combination, Title, Subject, Who (creator), What (subject), When (coverage), Where (coverage) and co-ordinate defined geographic areas. Thus a user might cross search the ADS and RCAHMS data sources for references to Roman (when) forts (what) in the border area between England and Scotland (user defined coordinates).

Although this specific instance of the Portal deals with the Historic Environment, it can be configured to deal with other domain-specific information. Furthermore, the framework implemented to support the Portal can be the basis for other services in order to implement a Digital Library.

The implementation of this framework is based on two purpose-designed Java packages, Zava, a Z39.50 API providing client/server features, and ZavaX, an Web Client API providing access to Zava from the Web. The system makes extensive use of XML and RDF for configuration and communication, and of XSLT for transformation and delivery of content.

This paper examines the technology, functionality, standards conformance and research potential of the Portal.

1 An Interoperability Solution

To support interoperability between diverse data sources, often with quite different internal schemas and stored data formats, a Portal needs to:

- offer a single interface to its users, irrespective of the number or type of query targets,
- translate the users' search criteria into semantically equivalent forms for each target, and
- collate and display search results when they become available.

*Sponsored by the Portuguese Fundação para a Ciência e a Tecnologia.

Compliance with metadata standards is a key factor in providing interoperability between different entities as it gives well-defined semantics at abstract levels allowing the necessary mappings. The Portal uses Dublin Core (DC) (Weibel, 1999; DCMI, 2000), Z39.50 (ANSI/NISO/ISO, 1995), and the Bath Profile (Lunau et al., 2000), a high level Z39.50 profile requiring DC and XML (Bray et al., 2000) as the basic standards. Using the Bath profile as a basis for interoperability and the Web for accessibility, a large number of resources can be made available to users through a single Portal. Users may apply high precision queries to any or all of the resources located on the connected data sources, subject to mediation according to the data source capabilities. These data sources are accessible transparently from the Web through the Portal which connects with a target located in each partner organisation containing access points to the real fields on the databases.

2 Portal Implementation

Two Java packages were developed to support the Portal infrastructure, Zava and ZavaX. Zava is a Z39.50 API offering basic client/server features, such as Init, Search, Present and Close. Internally, Zava uses an XML parser and a RDF (Lassila and Swick, 1998) processor to implement the profiles and to exchange the resources. ZavaX is based on a Web Client API allowing to process Z39.50 requests from the Web, supported by a Web Server powered by a Servlet Engine and by Zava. For searching it includes a Mediator component, and for retrieving, it includes an XML tag library (TagLib) application for Z39.50 supported by the Cocoon Publishing Framework (Apache XML Project, 2000), that associates specific XML markup with associated logic, providing a mapping between a set of XML tags and the Z39.50 parameters. It is used both to receive Z39.50 parameters embedded in XML documents from the Web, and to convert records obtained from the Z39.50 targets into XML records. Subsequently, these XML records can be transformed to any suitable format for the Web by using XSLT.

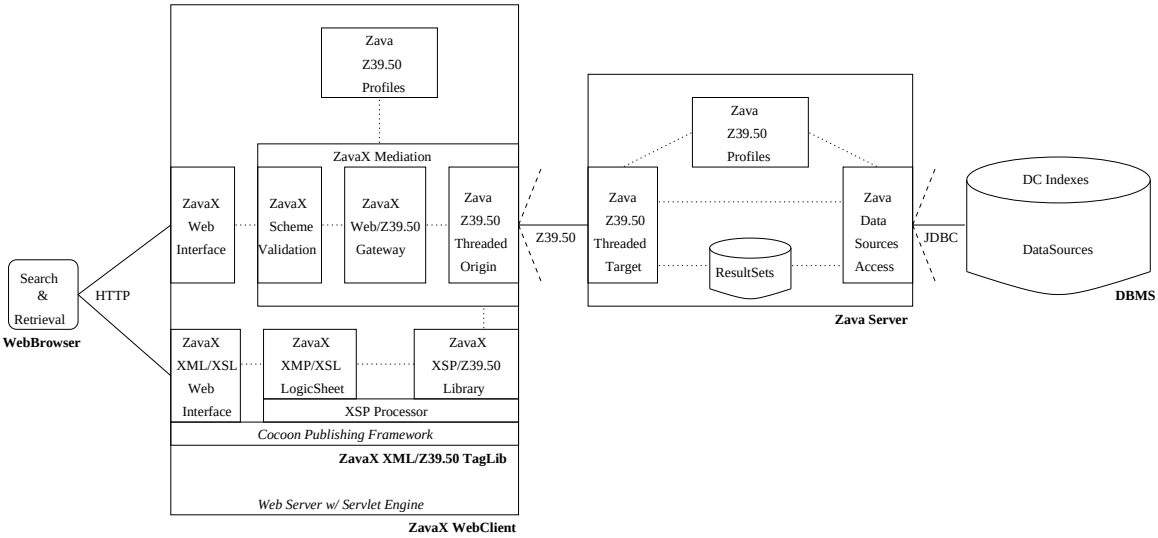

Figure 1: Portal Architecture.

The Portal Architecture is based on a 4-tier model providing uniform access from the Web to the data sources. The Portal front-end is a Web application implemented as a Java servlet, which can be accessed by HTML Forms or Applets, displayed with a Web Browser. Internally, the Portal is composed of a Mediator, Z39.50 Profiles and a XML/Z39.50 TagLib. The Mediator includes a Scheme Validator, a Web/Z39.50 Gateway and a Z39.50 Threaded Origin component. As shown in Figure 1, the Portal accepts search and retrieval parameters from the Web (e.g. host, port, database, query, query type, result set, element set, record

syntax), validates their values, and passes them to the Mediator. The Mediator selects the data sources according to their capabilities (e.g. Spatial Referencing System used) based on the profile configuration, converts the parameters to the Z39.50 protocol, launches parallel searches over all of the elected Z39.50 servers through their target access points, and finally presents the results as soon as they arrive using a TagLib.

Each Z39.50 server is composed of several components. A Z39.50 Threaded Target component supporting one or more profiles receives and processes Z39.50 connections. A profile component loads the profiles specified in RDF/XML in order to take decisions and obtain context about the information domain. Lastly, the Data Sources Access component supports access on one or more data sources, typically databases, via JDBC drivers that connect to TCP/IP listeners on each DBMS.

The Z39.50 servers expose one or more abstract databases providing access points for searching. Incoming search requests are mapped to the native queries (e.g. SQL) on the underlying physical databases. These physical databases contain indexes mapped onto DC metadata to implement the access points, and so provide access to their resources. Therefore, the Z39.50 servers work as wrappers hiding the heterogeneity of the different data sources and provide a uniform abstract databases to the Web Client. The result of each query generates result sets which are managed locally by the server and are used during the retrieval phase. Typically, a result set comprises a list of references to matching records, rather than copies of the records themselves. Each target may have different data sources and thus different access methods (e.g. JDBC, ODBC, private API). Additionally, each data source may have its own distinct schema. Having the same access points for the different schemas, one mapping is needed for each abstract database. Therefore, each target performs its respective mappings according to the abstract database being accessed, submits the native queries on their real databases, generates the result sets and returns the number of records found. As soon as the record counts arrive at the Portal, they are made available to the user who may then choose to retrieve the records from the target.

On the Portal, the records returned from the targets are always converted to XML for later processing by the TagLib, even if the target does not implement the optional XML record syntax. The TagLib transforms the XML records on-the-fly to the desired format (HTML, WML, PDF) and, according to the capabilities of the Web clients, may show different selected information (Browser in a PC or WAP client in a Mobile Phone).

Although Zava and ZavaX involve technologies usually not present on Z39.50 servers, the Portal is totally interoperable with any other Z39.50 Server. The only requirement for the server is to implement the same profiles as the Portal.

3 An Historic Environment Portal

The Historic Environment Portal aims to provide users with a better tool to find archaeological resources. Typical searching uses generic access points such as Title, Subject, Author or even Any (for searching several DC elements). As the Historic Environment emphasis is on spatial and temporal searching there are also Where and When access points. This is achieved using the DC.coverage element and feeding it with temporal and spatial descriptors based on thesauri of places and time period terms. Other access points include Who which allows searching by DC.creator or DC.publisher, and What which allows searching by DC.subject element of the resource.

Additional spatial access points are based on three different spatial referencing systems: OSGB (British National Grid), OSI (Irish National Grid) and LL (Latitude and Longitude geographical coordinates), and spatial co-ordinates for these referencing systems (SRS, X-Coord, Y-Coord). All of these can be combined into complex boolean expressions to perform

precise, effective and efficient searches. A possible query based on the generic access points on this Portal could be achieved taking a boolean option to search by Title=king, and Subject=fort at the selected targets. A more specialised query could be based on other access points such as What=fort and When=roman complemented with spatial co-ordinate access points based on the British or Irish national grids, as shown in Figure 2, or on geographical co-ordinates.

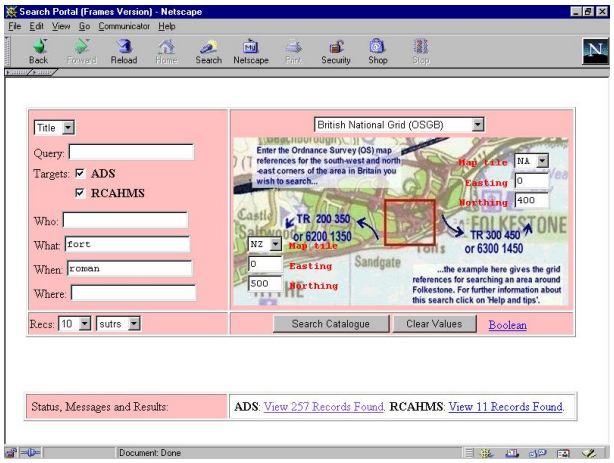

Figure 2: Prototype of the Portal Interface for Search.

Retrieval is based on result sets obtained during the search phase. In response to a present request, the target generates and retrieves records in a given record syntax format (e.g. XML, GRS1, SUTRS), which is always transformed to XML on the client side. At the Portal, the TagLib is then used to apply XSL Transformations to the returned XML records. In this way, the Portal can convert the XML records to a format that best suits the client application receiving the processed data. The users access the Portal normally through a Web browser, so the transformation will typically generate HTML. However, as the framework has the capacity to distinguish between different browsers and other clients, it can choose to deliver different content according to the capabilities of the client. If required, it could deliver appropriate content for a small handheld computer or even for a WAP enabled mobile phone, as shown in Figure 3.

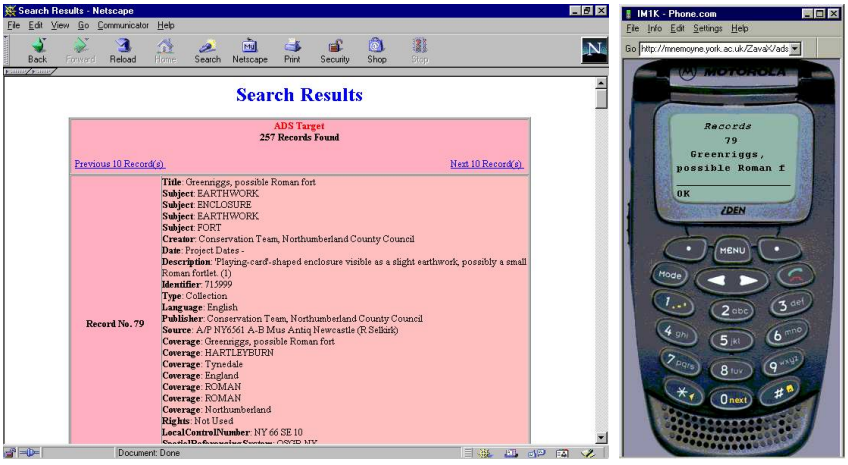

Figure 3: Record Converted to HTML in a Browser and to WML in a WAP Client.

Appropriate timeouts are used to ensure that the user does not wait indefinitely for a target that fails to respond to the initial search request (default 10 seconds), or fails to deliver results within a reasonable time (default 3 minutes). Other safeguards are built into the targets. To prevent large-scale data theft and to limit the load imposed on the target, the maximum

number of records in any query is limited and a message is returned suggesting that the user refines its query.

4 Conclusions

The main result obtained with this Portal is a solution to the interoperability problem of effective and efficient access to heterogeneous data sources applied to the Historic Environment information domain. Effective, because the access points have very precise and easily understood semantics, thus allowing users to search the data sources available under the targets and retrieve the records they are trying to find. The same results might be achieved by querying each data source using its own native query mechanism or interface, but only at a cost of considerable extra effort on the part of the user. Efficient, because the searches are launched in parallel over a set of selected targets according to a given criteria applied by a mediator and, even if some targets are down or busy, the Portal does not wait for the last answer before presenting the results. As soon as the targets finish their local searches and send their result counts, the records are available to be retrieved from their respective sources.

The main benefit for the end-user lies in the provision of a single, simple query interface from the Web to access complex heterogeneous distributed data sources. Rather than having to locate and understand many different sources of data, each with their own particular interface.

Additionally, by using the Bath profile ensures well-defined DC access points for searching and DC record elements encoded in XML for retrieving the records found from each target, allows us to have a common interface that can be used by any Bath compliant application. Therefore, the Portal facilitates the task of finding information as it concentrates access to all of the data sources in one place with a single interface.

Finally, the Portal connects the Web with Historic Environment or any other specific-domain data using DC, XML and Z39.50, thus ensuring that such sources can play a full part in the world of XML-based interoperability.

References

ANSI/NISO/ISO (1995). *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification (ANSI/NISO Z39.50-1995)/ISO 23950*. NISO Press, Bethesda, Md. http://lcweb.loc.gov/z3950/agency/.

Apache XML Project (2000). Cocoon publishing framework. *http://xml.apache.org/cocoon*.

Bray, T., Paoli, J., Sperberg-McQueen, C. M., and Maler, E. (2000). *Extensible Markup Language (XML) 1.0*. http://www.w3.org/TR/REC-xml.

DCMI (2000). Dublin core qualifiers. http://dublincore.org/documents/dces-qualifiers/.

Lassila, O. and Swick, R. R. (1998). *Resource Description Framework (RDF) Model and Syntax Specification*. http://www.w3.org/TR/WD-rdf-syntax/.

Lunau, C., Miller, P., and Moen, W. E. (2000). The bath profile: An international z39.50 specification for library applications and resource discovery. http://www.nlc-bnc.ca/bath/.

Weibel, S. (1999). The state of the dublin core metadata initiative. http://dublincore.org /documents/dces/.