

# A Prototype Implementation of a Framework for Organising Virtual Exhibitions over the Web

*Authors:*

*Ali Elbekai and Nick Rossiter*

**CIDOC06**  
**GOTHENBURG**  
**S W E D E N**

## 1 INTRODUCTION

As the Web becomes increasingly the world's official media, many museums, archives, libraries, and cultural heritage centres throughout the world invest in documenting their collections and in publishing their material via the web, making their information accessible on the Internet. Generally, museums use many different styles in presenting their collections on their Web pages. In addition, some museums provide sophisticated searching facilities so that the user can retrieve information about required items by subject, date and place. It is fairly certain that with such sophisticated new technologies, the need for standards to manage the information that these collections contain becomes more and more urgent. As a result, several projects have been conducted in the last few years in order to develop a unified standard for organizing and managing data and information in such institutions. In the same way, many museum organisations are working together to develop standards and techniques and to make their resources more widely and easily available. On this basis, (ICOM-CIDOC) and CIDOC documentation standard working group (DSWG) have engaged in the creation of a general data model for museums. CIDOC have particularly focused on information interchange. This effort resulted in 1999 in the first complete package of the "CIDOC Conceptual Reference Model" (CRM). Such an achievement was a result of intensive hard work from many experts in the field of IT and museum information, Many specialists contributed to this work, in order to fully exploit the potential of the CRM as a means to enable information interchange and integration in the museum community and beyond [ICOM-CIDOC 1996-2003].

As we know, data is very important to every industry: the way data is stored and transferred between applications, and the means to reduce the redundancy and inefficiency have gained in importance. A great deal of work in managing data has been done for e-commerce applications using EDI (Electronic Data Interchange). Currently work is being done on XML for data handling. In today's world, data presentation over the web is of concern. XML is one such technology that has created a revolution in handling and managing data [W3C 2004]. XML started out as a way of presenting data over the Web and has later been extended to a wide range

of applications because of its platform independent interoperable nature. The features such as data presentation over the web and interoperable nature have made XML popular in a wide range of applications. The coming of XML has widened the scope for companies and technologists the world over [W3C 2000, W3C 2004a]. It has enabled them to explore possibilities in the area of efficient data exchange, presentation of data and querying of the data. XML documents are self-describing and platform independent. Also the XML makes data exchange between different databases and different platforms relatively easy [Bourret 2004, Bourret 2005]. Furthermore XML is fast emerging as the dominant standard for representing and exchanging information between the web applications and different platforms where it is relatively suitable. So if data is stored and represented as XML documents, then it should be possible to query the contents and extract, present, transform and search data in a simpler way. To be more specific, XML (eXtensible Markup Language) is a markup language defined by the W3C that provides a way to create an extensible format for describing structured data in electronic documents and to express rules about this data [W3C 2004a].

This paper describes one further step in the direction of the accomplishment of truly pan-European collaboration for organisation of virtual exhibitions. Using a single XML schema for specification of the common exhibition information and utilising contemporary information technologies for processing XML data over the Web, the approach adopted demonstrates how to add a new European dimension to the inter-museum collaboration and to achieve wider access to the rich European cultural heritage. Furthermore the paper describes a prototype implementation of a framework for organising virtual exhibitions, which uses information provided by the collaborating museums in the form of Web services. The museum content published by the collaborating museums is organised into a homogeneous virtual exhibition space by an exhibition curator and is accessible from a single point of entry – the Virtual Exhibition site. The prototype assumes the content published is an extract of CIDOC-compliant museum database, which allows easy standardisation and further dissemination. The prototype system presented is built entirely using a public domain stack of technologies for processing XML data in Java (J2SE,

J2EE) and additional XML and Web Service packages. The rest of the paper is structured as follows. Section 2 presents the related work. Section 3 presents the prototype of the museum system. A subsection 3.1 introduces the design of the database of the system and subsection 3.2 presents an algorithm to generate an XSL stylesheet that will be used in the transformation layer. Section 4 leads to the architecture of the system. While section 5 presents the implementation part of the system, that is for exchange of information between several independent museums over the Web.

## 2 RELATED WORK

[Elbekai & Rossiter 2005b] presented a prototype of a framework for organising virtual exhibitions and describe one further step in the direction of accomplishment of truly pan-European collaboration for organisation of virtual exhibitions using a single XML Schema for specification of the common exhibition information and utilising contemporary information technologies for processing XML data over the Web. [Vassil Vassilev et al 1999] present both a general description and a technical specification of the information system for museum information processing. [Vassil Vassilev et al 2000] worked on previous projects for publishing museum content over the Web, in archaeology, in watermark images and in industrial heritage.

[Nicholas Crofts 2003] described a practical application of the CIDOC CRM in integrating a large and diverse set of data sources. These data sources, accumulated over a number of years, all contain information relating to Geneva's architectural and cultural heritage.

[IGMOI 1995, ICOM-CIDOC 1996-2003, CIDOC CRM 2004] that is being developed into an ISO standard as a joint effort of the CIDOC CRM SIG and ISO/TC46/SC4 (ISO/AWI 21127). CIDOC is the result of work done by the CIDOC Documentation Standards Group, from 1994-2000 and the CIDOC CRM Special Interest Group from 2000-2002, as the result of an initiative to define the underlying semantics of database schemata and document structures needed in museum documentation for the support of good practice in conceptual modeling, data transformation, data exchange, information integration and mediation of heterogeneous sources. [Bourret 2004] noted that XML and its related technologies have many facilities in common with real databases such as storage (XML documents), schemas (DTDs, XML schema languages) and

programming interfaces (SAX, DOM, and JDOM. [Wiederhold 1995] adopted a wrapper mediator architecture to provide a uniform user interface to query integrated views of heterogeneous information sources. [Manolescu et al 2001] described a query processor that works for different schema generation techniques. Their work is done in the context of data integration, and the tables generated by each relational schema generation technique are specified as materialized views over a virtual global schema. [Elbekai & Rossiter 2005a] presented an algorithm as a technological solution for XQuery interpreter generating. Also, we presented an XSL stylesheet that can be used for transforming XML query to SQL query and in addition we will use the XSL stylesheet in the implementation of a framework for organising virtual exhibitions. The work described here is based on the previous paper we have presented in the CIDOC 2005 [Elbekai & Rossiter 2005b].

### 3 A PROTOTYPE OF THE MUSEUM SYSTEM

This section will introduce a prototype implementation of a framework for organising virtual exhibitions over the Web. The prototype assumes the content published is an extract of CIDOC-compliant museum database, which allows easy standardisation and further dissemination. The prototype system presented is built entirely using public domain stack of technologies for processing XML data in Java (J2SE, J2EE and additional XML and Web Services packages). It functions as an entirely server-side Web application executed by Tomcat server connected to a backend database (one for each participating museum).

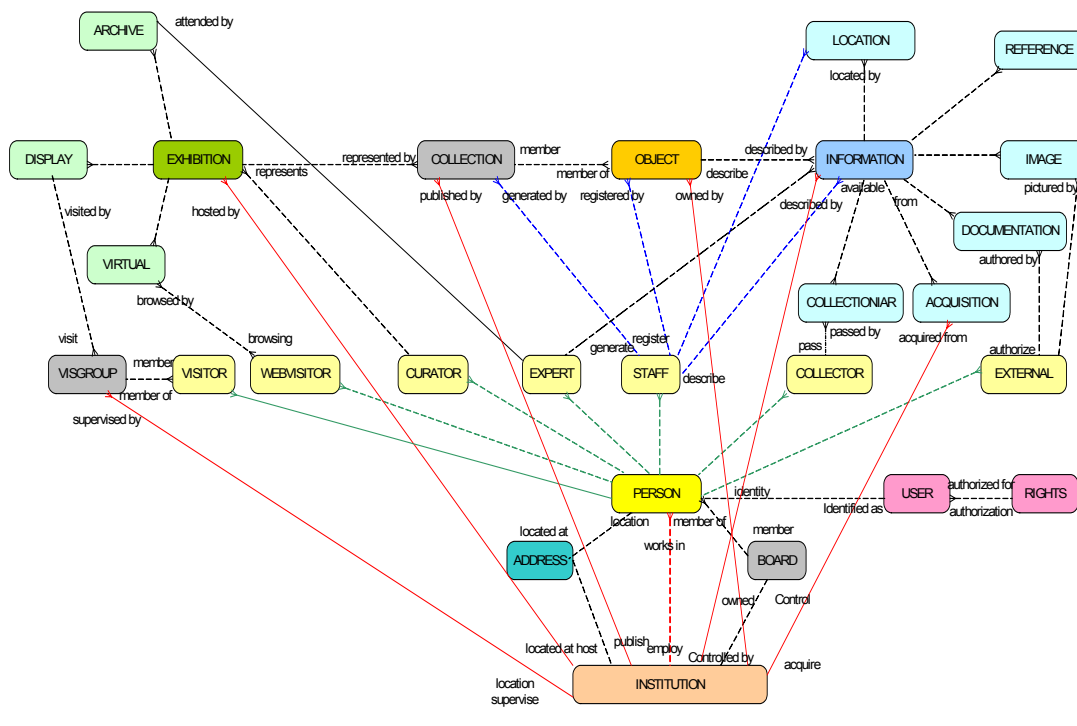
#### 3.1 Design of Database for the proposed system

Entity Relationship Diagrams [Chen P. P. 1976] provide the logical structure of the databases for our proposed system. Figure 1 shows the cardinality constraints between the objects classes. As can be seen from the diagram each Collection may relate to a number of Objects. Also the Object may contain much Information such as Location, Reference, Image, Documentation, Acquisition and Collection. The Exhibition has many Collections, and the Exhibition can be of different kinds such as a public display, virtual exhibition or archive (type of exhibition). An Institution

has many Visitor groups, Exhibitions, Collections, Objects, Information and one Address.

Note that the Person could be in a one to many constraint with Staff, Expert, Curator, Web visitor, a Visitor group, Collector or External. More details on the entity relationships between the objects classes for our proposed system are shown in *Figure 1*.

Figure 1: Entity Relationship Diagrams for the museum system



### 3.2 An algorithm for generating XSL stylesheet

In this section we introduce an algorithm for generating an XSL stylesheet from the XMLSchema to interpret the XQuery as shown in *Figure 2*. In other words, we use this algorithm for generating the XSL stylesheet and then will use the generated XSL stylesheet shown in *Figure 3* for transforming XQueries to SQL queries and then in the implementation of a framework for organising virtual exhibitions as we have installed the XSL in the museum server [Elbekai & Rossiter 2005b].

Figure 2: An algorithm for generating XSL to transform XQueries to SQL queries.

```

// Input XMLSchema, XSL stylesheet
// Output XSL stylesheet (Generic XSL stylesheet)
1. start
2. if the input arguments (XMLSchema, XSL stylesheet) exist
    2.1 Building DOM and parsing it
    2.2 if parsing XMLSchema is done, then DOM will build dynamically
        2.2.1 perform XSL stylesheet transformation
        2.2.2 if transformation is done
            2.2.2.1 XSL templates start from the root node of DOM tree
            2.2.2.2 If the root node has parent/child node
                2.2.2.2.1 the XSL stylesheet template will walk and matches nodes, if the template matches
                    IN node in DOM tree nodes
                2.2.2.2.2 the XSL template adds FROM clauses to the new template and mapped the
                    root node of DOM to the (new template in new XSL stylesheet)
                2.2.2.2.3 the XSL stylesheet that pulls the nodes from DOM tree and places it with
                    formatting, into a new XSL stylesheet
                2.2.2.2.4 the XSL stylesheet template walks through the DOM tree nodes and when the
                    template matches RETURN
                2.2.2.2.5 the XSL template will add the SELECT clauses to the template
                2.2.2.2.6 map the parent/child of DOM tree as new template to the new XSL stylesheet
                2.2.2.2.7 iterate and walk through the DOM tree nodes and when the XSL template matches
                    nodes has name WHERE or IF or Else clause
                2.2.2.2.8 set the WHERE clauses as a new template into the new XSL stylesheet
                2.2.2.2.9 When all the templates have been executed and placed in the output
                2.2.2.2.10 then the new template will be a generic XSL stylesheet
            2.2.2.3 report no parent/child node and terminate
        2.2.3 report transformation errors and terminate
    2.3 report parsing errors and terminate
3. report arguments error and terminate
4. terminate/end

```

Figure 3: Generic XSL Stylesheet (XQuerytoSQLTrans.xsl) for transforming XQuery to SQL

```

C:\Tomcat\webapps\xml\WEB-INF\classes>jx GenerateXSLXQuerySQL GenXQueryXSL.xsl XMLSchema.xsd
C:\Tomcat\webapps\xml\WEB-INF\classes>c:\jdk1.4.0\bin\java -classpath c:\oracle\ora92\jdbc\lib\
sses12.jar;. GenerateXSLXQuerySQL GenXQueryXSL.xsl XMLSchema.xsd

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="text"/>
<xsl:template match="return">
    SELECT <xsl:value-of select="@name"/>
</xsl:template>
<xsl:template match="in">
    FROM <xsl:value-of select="@name"/>
</xsl:template>
<xsl:template match="Where">
    WHERE <xsl:value-of select="@name"/>
</xsl:template>
<xsl:template match="equal">
    FROM <xsl:value-of select="@name"/>
</xsl:template>
<xsl:template match="if">
    WHERE <xsl:value-of select="@name"/>
</xsl:template>
<xsl:template match="then">
    SELECT <xsl:value-of select="@name"/>
</xsl:template>
<xsl:template match="else">
<xsl:value-of select="@name"/>
</xsl:template>
<xsl:template mode="@name" match="in">
<xsl:call-template name="lower">
<xsl:with-param select="@name" name="value"/>
</xsl:call-template>
</xsl:template>
<xsl:template mode="@name" match="Where">
<xsl:call-template name="lower">
<xsl:with-param select="@name" name="value"/>
</xsl:call-template>
</xsl:template>
<xsl:template mode="@name" match="equal">
<xsl:call-template name="lower">
<xsl:with-param select="@name" name="value"/>

```

## 4 ARCHITECTURE OF THE MUSEUM SYSTEM

The architecture as shown in *Figure 4* is composed of three components as follows:

Web browser (Client) that can connect to the server, i.e. to access the Java servlets, the client can use PCs to run a Java servlet.

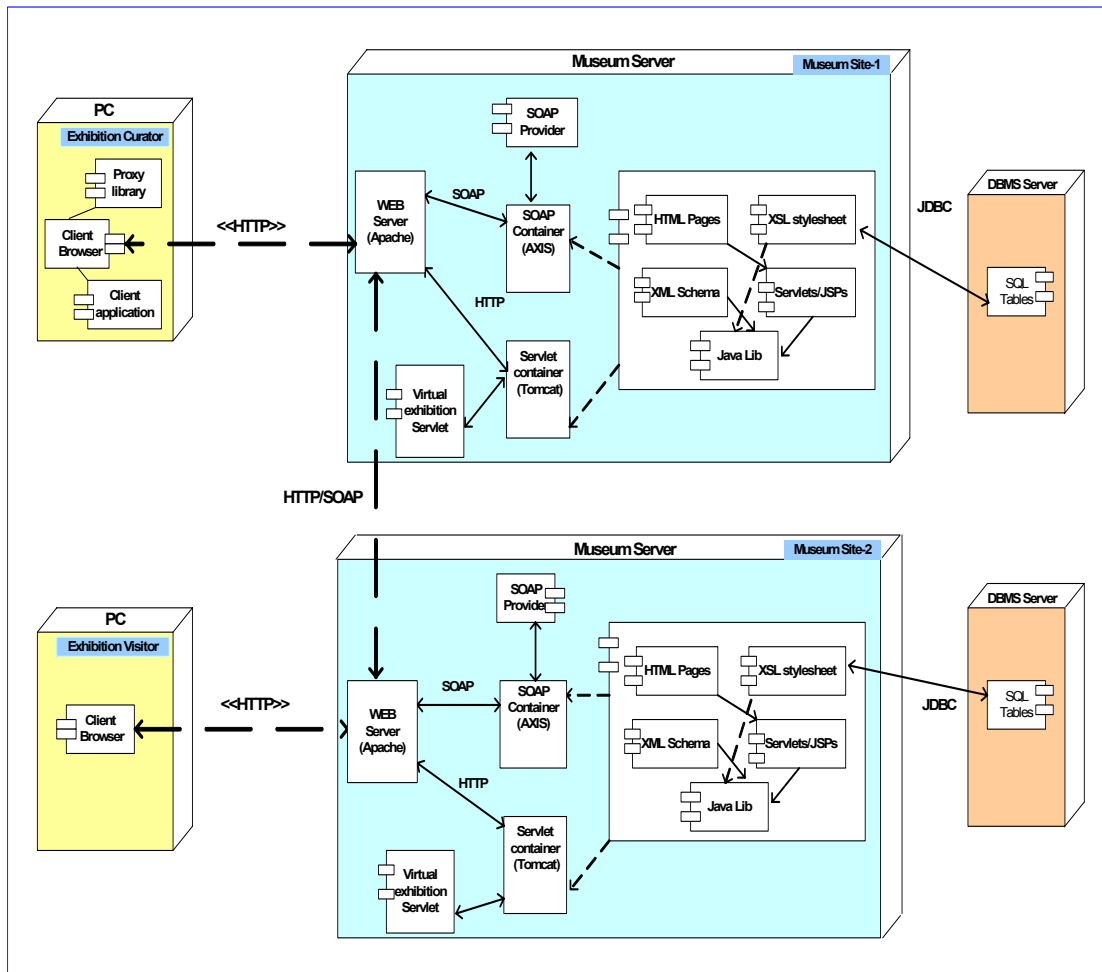
Middle server, i.e. a set of servers and internal network connecting them. This provides a web server capable of accessing data from DBMS and making it available to the client. Technology choices for the middle-tier include a Web server, Web sever with servlets (Tomcat), SOAP container (AXIS), a Virtual exhibition servlet, Java Server Pages, HTML Pages and XSL stylesheets [W3C 2001]. The communication protocol between the database and the museum server could be JDBC.

DBMS server with SQL tables provides database storage.

The following subsection will introduce an example to evaluate the integrated approach and show how the user can formulate an XQuery [W3C 2004b] and send it to the web server over the HTTP protocol. Then the XSL Libraries will transform the XQuery to SQL and the JDBC will contact the database and store the data.



Figure 4: Architecture the museum system



## 5. IMPLEMENTATION OF THE SYSTEM

This part describes the implementation of the system. In the following sections the processes performed by the algorithm presented in section 3.2 and the prototype will be illustrated by various screens.

### 5.1 An Example for display information

The client browser is able to display institutional information. The HTTP client sends an HTTP GET parameters request such as a XQuery For inst in <institution> Return <inst> for display institution information to a Web server, then sends a request to the servlet container tomcat (Step1). The Java servlet will use the XSL stylesheet (XQuerytoSQLTran.xml) on the server to

transform the XQuery user request to SQL query string (Step 2). Once the transformation is done, the SQL query string is generated and then the Java servlet connects to DB and passes the SQL query string to the database server over JDBC (Step 3). The Java servlet class will retrieve the information according to the SQL query string. Then the XSL stylesheet will transform the retrieved data to XML and send the output back to the client (Step 4). The result is shown in *Figure 5* and again on Tomcat server in *Figure 6*.

Figure 5 Institutional information displayed

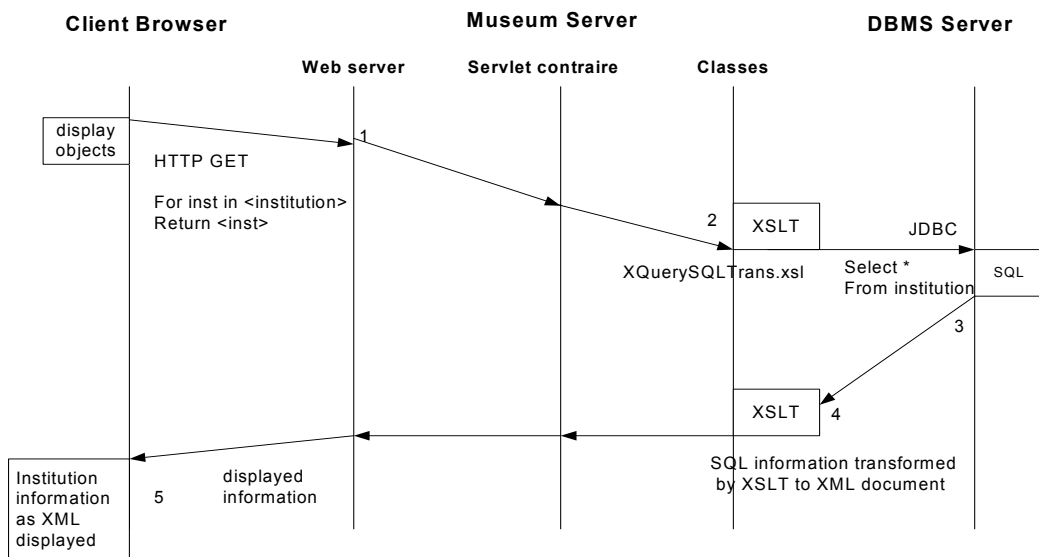
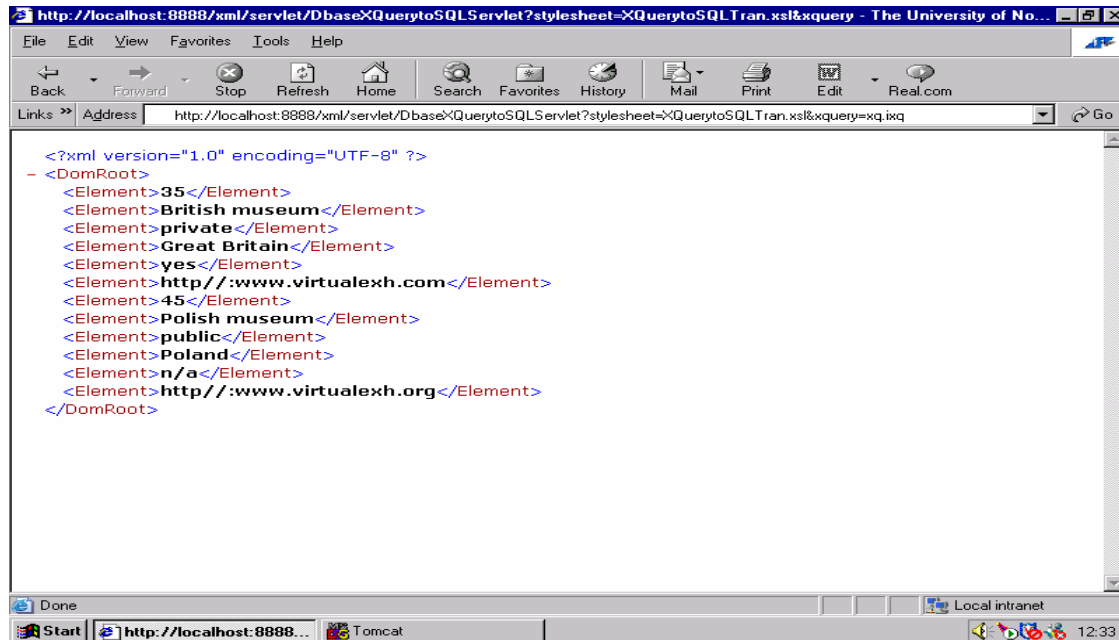


Figure 6 Institutional Information displayed on Tomcat server



## 5.2 An example for search and display of information

After a successful login the client is able to search and display object information. The client specifies and sends an XQuery such as `For obj in <collection> Return <obj>` Where `obj/<regis_Date> > 10.01.02` to the Web server as a URL by using HTTP (Step1). The museum server parses and transforms the request by the XSL stylesheet (`XQuerytoSQLTran.xml`) on the server and creates a SQL query string (Step 2). Once the transformation is done and the SQL query string generated, then the Java servlet is connected to the DB server and the SQL query passed to the database server over JDBC (Step 3). The query is executed. Then the database server returns the report to the museum server.

The XSL stylesheet will transform the report to HTML by using the XSL stylesheet (`GenXSLHTML.xml`) and sending the output back to the client (Step 4). This result is shown in *Figure 7* and also on Tomcat server *Figure 8*.

Figure 7: Search and display objects

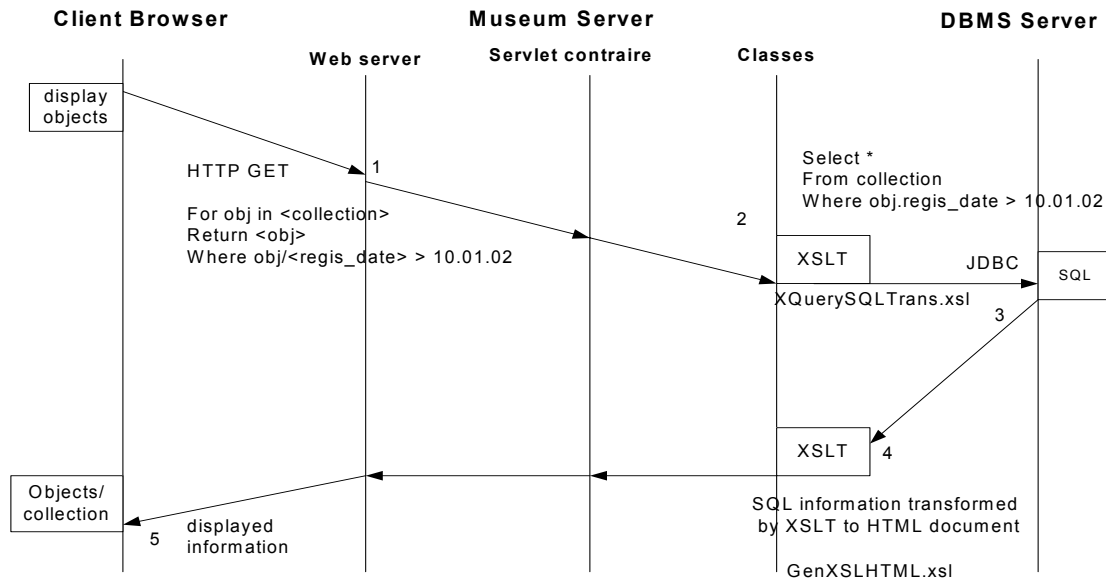
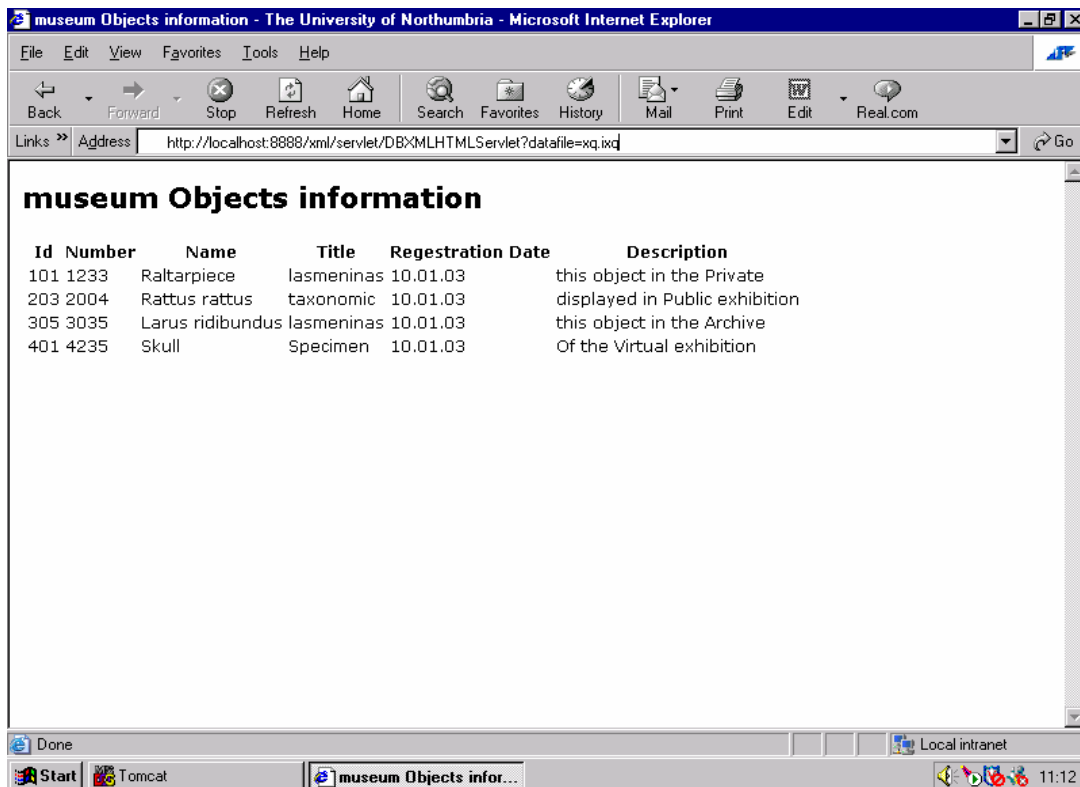


Figure 8: Objects displayed by the Client on Tomcat Server



### 5.3 An example for search and browse information

The client sends an HTTP GET request with the XQuery such as For obj in <collection> Return <object> Where obj/<regis\_date>=10.01.03 to museum server to display the objects that have registrations date equal to 10.01.03 (Step1). The XSL stylesheet (XQuerytoSQLTran.xml) generated in Chapter 5 will transform the XQuery to the SQL query string (Step 2). Once the transformation is done and the SQL query string generated, then the Java servlet is connected to the DB server and the SQL query passed to the database server with JDBC (Step 3). Now the Java servlet retrieves the information according to the SQL query string. Again the XSLT will transform the client request to HTML by using the XSL stylesheet (GenXSLHTML.xml) and sends the output back to the client (Step 4). This result is shown in *Figure 9* and again on Tomcat Servlet in *Figure 10*.

Figure 9: Search and display information

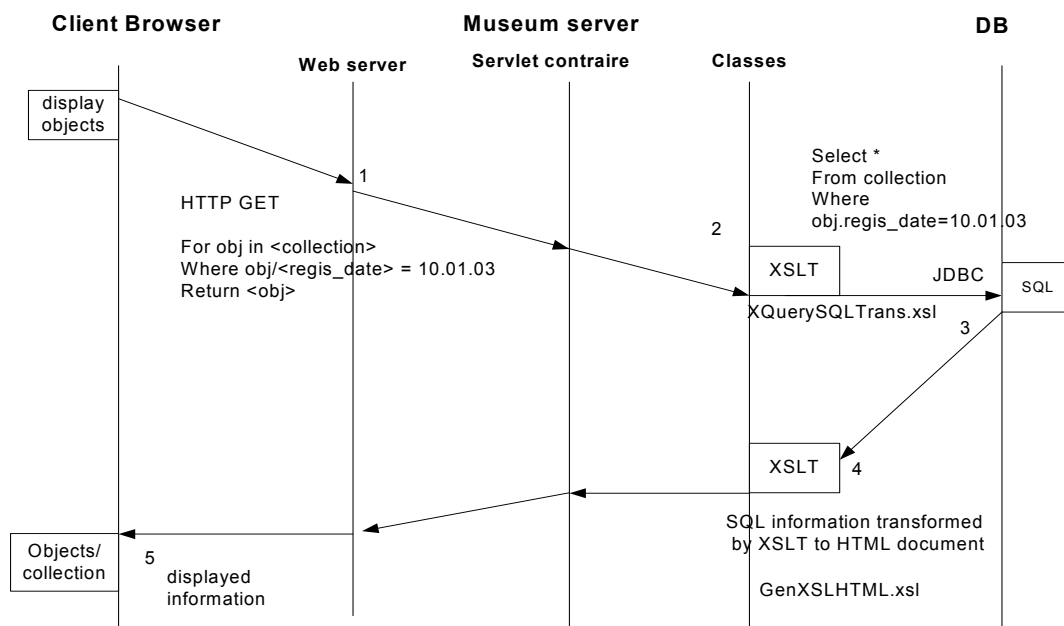


Figure 10: Objects with registration date is 10.01.03 on Tomcat Server

Id	Number	Name	Title	Registration Date	Description
101	1233	Raltarpiece	lasmeninas	10.01.03	this object in the Private
203	2004	Rattus rattus	taxonomic	10.01.03	displayed in Public exhibition
305	3035	Larus ridibundus	lasmeninas	10.01.03	this object in the Archive
401	4235	Skull	Specimen	10.01.03	Of the Virtual exhibition

#### 5.4 An example for search on an exhibition information

After a successful login the client is able to search and display exhibition information. The client specifies a XQuery like (For exh in <exhibition> Return <exh>) Where exh/<opened> = “yes” to Web browser as a URL by an HTTP request (Step 1). The Web browser sends the URL to the data server by Java servlet (Step 2). Then the museum server parses the URL request and transforms the XQuery to an SQL query string by using the XSL stylesheet (XQuerytoSQLTran.xml) (Step 3). The museum server passes the SQL query to the database server over JDBC (Step 4). Then the SQL query is executed and the result produced (Step 5). Again the XSLT will transform the result to HTML by using the XSL stylesheet (GenXSLHTML.xml) and sends it to the Web server (Step 6), then the Web server sends the output back to the client (Step 7). This is shown in *Figure 11* and on Tomcat server in *Figure 12*.

Figure 11: Search and display information

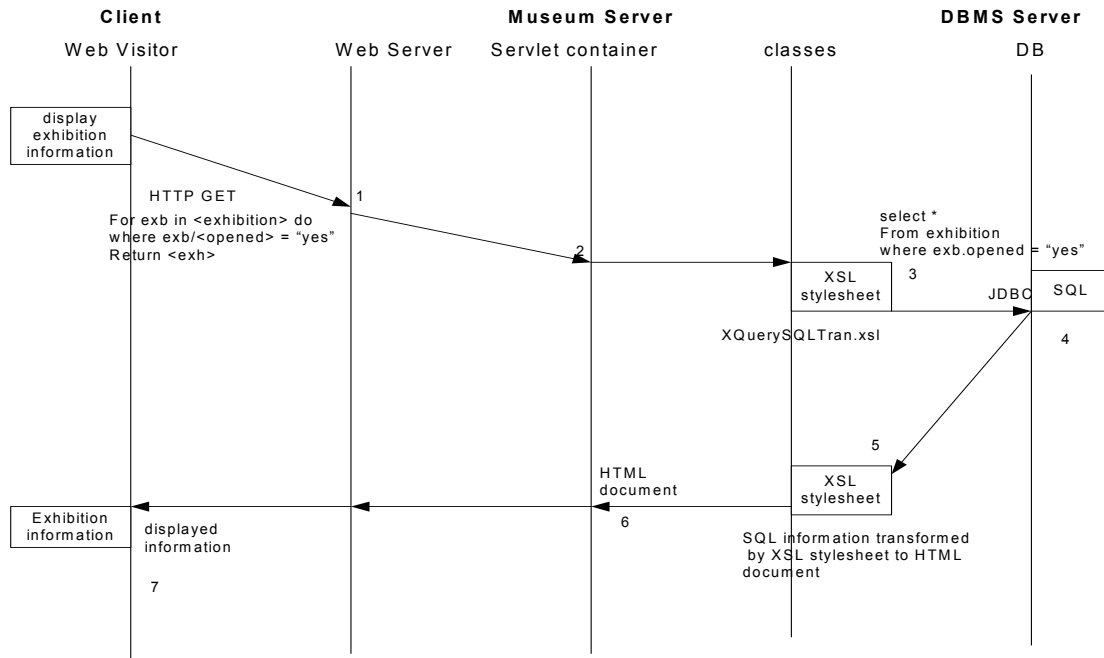
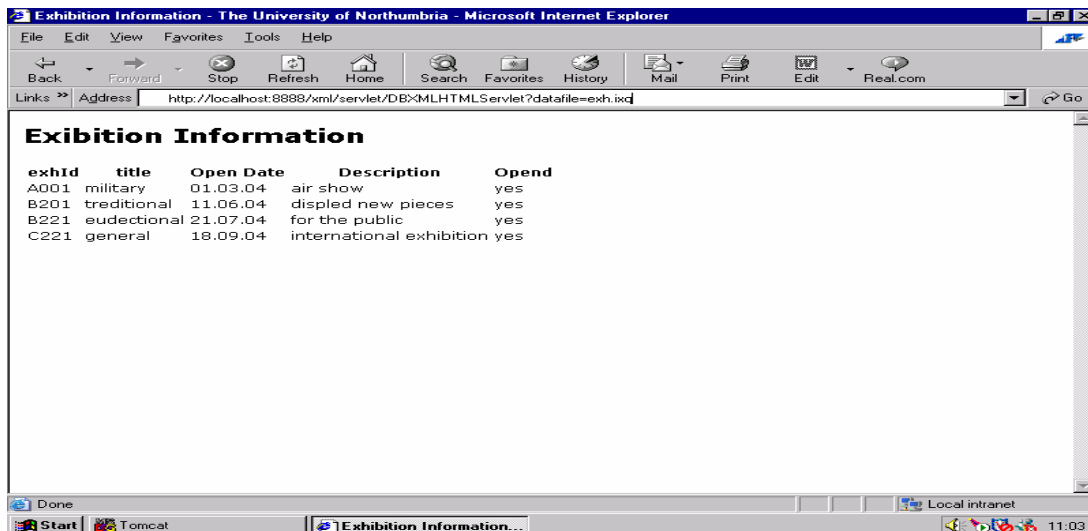


Figure 12: Opening exhibition information on Tomcat Server



## 6 CONCLUSION

The contribution of this work is it that introduces a prototype implementation of a framework for organising virtual exhibitions, which uses information provided by the collaborating museums in the form of Web services. The museum content published by the collaborating museums is organised in a homogeneous virtual exhibition space by an exhibition curator and accessible from a single point of entry - the Virtual Exhibition site. The prototype assumes the content published is an extract of a CIDOC-compliant museum database, which allows easy standardisation and further dissemination. The prototype implementation of the system presented is built entirely using public domain stack of technologies for processing XML data in Java (J2SE, J2EE and additional XML and Web Services packages). It functions as an entirely server-side Web application, executed by Tomcat server connected to a backend database. Furthermore, this paper describes one further step in the direction of accomplishment of truly pan-European collaboration for organisation of virtual exhibitions using a single XML Schema for specification of the common exhibition information and utilising contemporary information technologies for processing XML data over the Web. We have performed some test examples by selecting certain screens which were mainly based on XML schema. A further aim of these examples is to show a better practical system for exchange of information between several independent museums over the Web. Furthermore, we have presented and used an XSL stylesheet for interpreting the XML query to SQL and in the implementation of a framework for organising virtual exhibitions.



## 7 REFERENCES

- Bourret Ronald*. XML and Database. Published online at <http://www.rpbouret.com/xml/XMLAndDatabases.htm>, last updated July (2004).
- Bourret R.* XML and Database. <http://www.rpbouret.com/xml/XMLAndDatabases.htm>. (2005).
- Chen P. P.* The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*. 1(1):9-39. (1976).
- CIDOC CRM*. Mappings to the CIDOC CRM can be accessed from a Technical Papers list on the CIDOC CRM SIG Web site, at [http://cidoc.ics.forth.gr/technical\\_papers.html](http://cidoc.ics.forth.gr/technical_papers.html), April (2004).
- Crofts, Nicholas*. Combining data sources – prototype applications developed for Geneva's department of historical sites and monuments based on the CIDOC CRM (2003).
- Elbekai Ali & Rossiter Nick*. XML Schema-Driven Generation of Architecture Components, Workshop on Web Services: Modelling, Architecture and Infrastructure, 7th ICEIS, Florida, 159-164 (2005a).
- Elbekai Ali & Rossiter Nick*. Virtual Exhibitions Framework: Utilisation of XML Data Processing for Sharing Museum Content over the Web, Proc CIDOC Annual Conference 2005, Zagreb, Croatia, 24-27 May, CD-ROM, paper 7, 16pp (2005b).
- ICOM-CIDOC*. Introduction to the International Committee for Documentation of the International Council of Museums ICOM-CIDOC (1996-2003).
- IGMOI*: International Guidelines for Museum Object Information: The CIDOC Information Categories, produced by the (CIDOC) of the (ICOM), October (1995). <http://www.cidoc.icom.org/guide/guide.html>.
- Manolescu I., D. Florescu, D. Kossman*, "Answering XML Queries over Heterogeneous Data Sources". 27<sup>th</sup> VLDB Conference, pp. 241- 250, Rome, Italy, September (2001).
- Vassilev V., I. Stoev, I. Rangelova et al.* Museum Information Systems: CIDOC data model implementation in the ArchTerra project. *Bol. CILEA No. 69* (1999).
- Vassilev V., I. Rangelova, G. Simeonova et al.* Drill-down Navigation inside Archaeological Museum Database. Dynamic Classification and Controlled Terminology Implementation. Proc. EAA2000, Portugal Institute of Archaeology, Lisboa, Portugal (2000).
- W3C Extensible Markup Language (XML) 1.0 (second Edition), W3C Recommendation, <http://www.w3c.org/TR/REC-xml>. (2000).
- W3C. eXtensible Stylesheet Language Version 1.0, W3C Candidate Recommendation, October 15, 2001. <http://www.w3.org/TR/2001/REC-xsl-20011015/> (2001).
- W3C, "XML Information set", [www.w3.org/TR/xml-infoset/](http://www.w3.org/TR/xml-infoset/): last version February (2004a).
- W3C. XQuery 1.0: An XML Query Language, W3C Working Draft July 23 (2004b).
- Wiederhold, G. Mediation in Information Systems. *ACM Comput. Surv.* 27(2) 265-267 (1995).