An Efficient Distributed Model for XMLised Blob Data Generation

Lavanya Rajendran
Department of Media Sciences,
CEG, Anna University,
Chennai

Ramachandran Veilumuthu
Department of Information Science and Technology,
CEG, Anna University
Chennai

ABSTRACT

In today's Internet computing, eXtensible Markup Language (XML) is preferred tool for exchange of data's by all the organizations irrespective of its size. As there are lots of data exchange is involved within the organization as well as outside the organization, there is high requirement for reliable, consistent, and platform-independent mode of data sharing especially with respect to multimedia files like audio and video files. The major problem with most of the relational database is its incompatible formats between computer systems. XML will be hardware and software independent thereby having full integration with any environments. Thus, this paper aims to convert or transform the multimedia files stored in the relational database into an xml document for data sharing in a distributed Environment. As XML is text based, data sharing becomes very easy and faster when compared to the multimedia files, thereby increasing the internet computing performance. By providing proper authentication details, the blob data stored in relational database can be fetched in a XML format that can be used independently for multiple purpose.

Keywords: Relational Database, XML Generation, Data Sharing, Distributed Environment.

1. INTRODUCTION

"The Database is a collection of large, integrated data. The Relational DataBase Management System (RDBMS) is software that stores, maintains and manages the data's in it. There are few dbms available that even secures the data effectively. In the current scenario, when the data's stored in the database are shared, people face lots of problems like duplicating, altering and utilization. The same efficient and secure method is required when transforming the data in a distributed environment, especially when the size of data is large as in case of binary large objects (blob) data's. The other major problem with blobs is that, the universal database support for this data type is very minimum.

In the distributed environment, the problems like interoperability issues between different operating systems and the sharing of blob data from one end to another in an understandable mode has to be addressed. Unlike, RDBMS which is in a machine readable format, XML document is easily human readable and is self-explanatory. It is also very simple to add or remove data to a node in an xml document, by still keeping the size of the file in manageable state.

For the data sharing and exchanging, using XML is widely accepted as a standard. Therefore, this paper, uses the JAX-WS, a java web service model to convert the blob data's from the database to xml format and convert the same as a "XMLisedBlobDataGeneration" service in a distributed environment. The XML Blob data representation is not only better to share the contents but it is also easier for querying the required data.

The research paper is divided into five main sessions. The first session focus on the works done related to adopting different methodologies to convert the database to XML format. Followed which the benefits of Distributed Environment and the proposed web service architecture are discussed. The third session describes in detail, the storing and retrieval of video from the relational database and the implementation of converting the blob data into xml format. Finally the paper concludes the importance of using xml to transfer the blob data in the distributed environment.

2. RELATED WORKS

In 2001, Joseph and Francis has made an attempt where the semantics of the relational database was captured in EER diagram and the same are mapped to XML schema using stepwise procedures. The physical data are then mapped to XML document under the definitions of the XML schema[1]. As this was one way conversion in the year 2002, the bi-directional conversion between database management systems and xml document was developed and presented as a new XML language namely DBXML[4]. The DBXML system has a simple GUI (Graphical User Interface) that works on few of the previous versions of Windows. This lacked to support other operating systems, platforms and RDBMS.

Another research formulated a new Xtables architecture, where users with single xml query language can create and query xml views of relational data's[3]. In most of the above cases, the xml enabled database management systems can translate only a few relations into an xml document. In 2004, an attempt to replicate XML database for a company's RDB to improve database performance, and to automate XML database recovery in case of system failures[2]. It also ensures that for any successful update to the RDB, corresponding update will be applied to its replicate XML database. Similarly there were many research that had concentrated on converting the relational database to xml document.

Even Recently in 2009, to achieve the urban traffic control system in easy way to share the data, the advantages of XML data model from the point of data exchanging standardization and the design flow of traffic control markup languages were discussed[5]. As XML-based message formats are already widely adopted in other IT domains such as the Web and would be perfectly suited for data exchange in heterogeneous environments including the embedded sensor actor networks[8]. In the above cases, only the raw text data has been used for conversion from the database to the xml document and vice-versa. There was no attempt in converting the Blob data's that are stored in the relational database into an xml format, and illustrates the reason for converswhy the study is important to adopt in distributed environment.

Thus, this paper aims to convert the multimedia files like audio and video files which are stored as blobs in relational database to xml format. As this conversion is a web service based, communication between multiple computers can be widely used for faster data sharing within and outside the organization. This proposed model also ensures basic security as it requires proper authentication for the conversion.

3. DISTRIBUTED ENVIRONMENT

In the current scenario, it is very essential to develop an integrated and distributed application to adopt the usefulness of the service to its fullest. Thus the stand-alone application of converting of blob data's to xml document is transformed as a web service that can be accessed by clients in the distributed environment. The Java API for XML (JAX-WS) is used for building the XMLisedBlobDataGeneration as a web service. By this method, the XMLisedBlobDataGeneration service call is also represented by SOAP protocol that is also XML based. Due to the inherent property of "JAX-WS", the XMLisedBlobDataGeneration service also becomes platform independent and it can be invoked or shared by or used as a components of distributed, web-based application from any platform. The WSDL file is also an interface that is available in the machine-processable format for the client to utilize the service effectively. The servlet deployed in the server fetches the video blob data from the relational database, which is been utilized by the XMLGeneration.java to convert it to the xml format. The architecture of the developed model which is converted to a web service is shown in Figure 1.

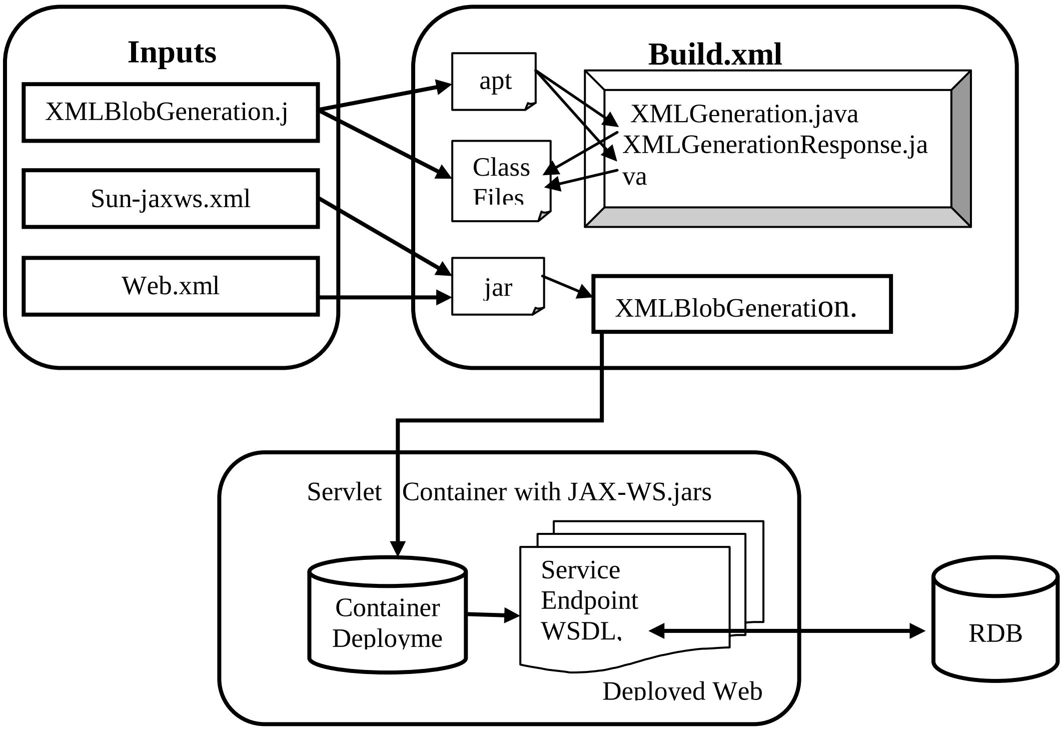

Figure 1 : Architecture of the proposed XMLisedBlobGenerationService

4. VIDEO STORAGE AND RETRIEVAL

The Java DataBase Connectivity Api is used to insert and retrieve the data's including video files from the relational database. The Structured Query Language is used as the query language. The insertion of videos is done through the prepared statements. The video file is first converted to byte array and then inserted to the table through prepareStatement method. Similarly the fetching of blob data is executed through SQL queries. The following Figure 2 shows the control flow of how the queries are executed.

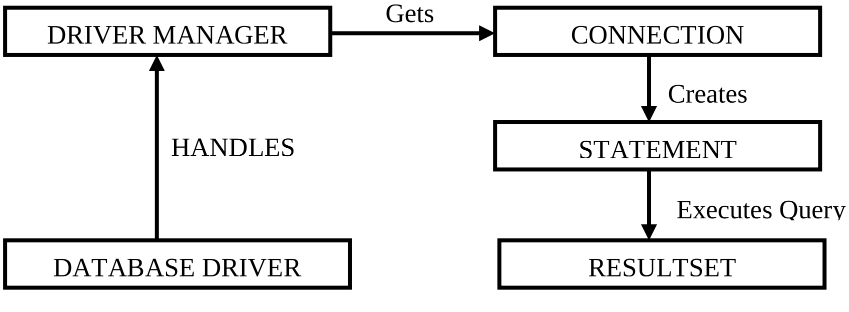

Figure 2: Control Flow for SQL query execution

5. WEB SERVICE IMPLEMENTATION

The following code shows the implementation of the web service XMLBlobGeneration that consists of the XMLGeneration method, that accepts xmlFileName as it Parameter when the service is invoked and returns the fetched video as string. It also creates a XML file in the Client's machine that contains all the fetched data in the xml format for processing.

```
@WebService()
public class XMLBlobGeneration {
… … …
   @WebMethod(operationName = "XMLGeneration")
   public String XMLGeneration(@WebParam(name =
      "xmlFileName")
   String xmlFileName) {
            … … …
   return success;
   }
}
```

The Endpoint configuration of the proposed service is shown below:

```
<endpoint
      implementation="org.me.BlobGeneration.XMLBlobGen
      eration"
               name="XMLBlobGeneration"            url-
      pattern="/XMLBlobGeneration"/>
```

The configuration of the web service client, which consists of the WSDL location is shown below:

```
WebServiceClient(name="XMLBlobGenerationService",
      targetNamespace="http://BlobGeneration.me.org/",
      wsdlLocation="http://localhost:8000/XMLBlobGeneratio
      nWSAppln/XMLBlobGeneration?wsdl")
```

The remote client invoking the XMLGeneration method that is deployed in the remote server. The returned string is stored in the string variable result.

```
org.me.blobgeneration.XMLBlobGenerationService service =
      new
org.me.blobgeneration.XMLBlobGenerationService();
org.me.blobgeneration.XMLBlobGeneration port =
service.getXMLBlobGenerationPort();
java.lang.String xmlFileName = filename;
java.lang.String result = port.xmlGeneration(xmlFileName);
```

5.1 eXtensible Markup Language

XML uses the non-proprietary file format (plain text file) with standard mark-up specification, thereby allowing easy and faster portability. XML format allows the systems, components and websites to communicate in a common Language. Hence, it is always preferred to send the data in the form of XML which contains only the data and not the presentation. Instead of sending the whole database, an attempt as been made to XMLise the table data's that is stored in relational database. As XML is a platform-independent, self-describing language, it is used to exchange the table data's in an expandable way. This generated XML file after converting to web services can be used where ever necessary in the distributed environment. This way the table data's are stored in a different form (textual) that can be customized depending upon the need. The requirement for processing of this distributed model can be achieved by multiple distributed physical computers. As this model is developed using JAX-WS, the implementation of the procedure call for converting the relational database to xml is performed in a remote system.

5.2 Xml Scheme for the Representing the Video Blob

The video blobs are converted into a simple XML format which will be clearly understandable and suitable for the clients. To provide a fault tolerant web service, the XML document has to be enhanced and validated without any data format error. The XML scheme or the Data Type Declaration is used to format a XML document. The scheme defines the exact structure in a descriptive form to store the entire video blob and its meta data's. The DTD for the XML document which stores the video data's from the relational database is shown below:

```
<!DOCTYPE moviedb [
<!ELEMENT moviedb (movie+) >
<!ELEMENT movie(moviename?, movieintext?)>
<!ELEMENT moviename (#PCDATA)>
<!ELEMENT movieintext (#PCDATA)>
]>
```

The root element of the xml document is moviedb that contains child element movie. The element movie is mandatory and appears either once or indefinitely. It contains two child elements namely moviename and the movieintext, which appears once or never as a child element. The data inside moviename and movieintext are movie name and the video converted to text respectively.

5.3 Data Fetching

The client sends the video name and the xml file name as parameters to the server, which inturns converts the video into xml format and stores the same in the client's machine with the file name specified by the client. The sample data from the xml document that contains the video blob data's is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<moviedbxml>
  <moviedetail>
        <moviename> </moviename>
<movie>AAAAHGZ0eXBtcDQyAAAAAGlzb21hdmMxbX
A0MgAAOCttb292AAAAbG12aGQAAAAAxaJN+sWiTfoA
AAJYAABi5wABAAABAAAAAAAAAAAAAAAAAQAA
AAAAAAAAAAAAAAAAAAEAAAAAAAAAAAAAAA
AAAEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAADAAAAFWlvZHMAAAA
             AEAcATKRXAAAfgXRy  ……   ….  …..   ….
….  ZFkAwZwAMPgA==
        </movie>
   </moviedetail>
</moviedbxml>
```

The xml document stored in the clients machine can be converted to any format like html, pdf or word document as required by the user using the XML Stylesheet Language (XSL). It can also be converted back to video format to watch the video.

6. CONCLUSION

This research paper proposes the use of XML, which is very easy to integrate with the web applications for data exchange of video blobs. As XML consists of only text, it is easy to transmit through any protocol and portable between different platforms. Thus, this paper converts the video blob from relational database into xml format which is then transformed into web services for using it in a distributed environment.

7. REFERENCES

[1] Fong, J. Pang, F. Bloor, C., "Converting relational database into XML document". IEEE Conference Proceedings in Database and Expert Systems Applications, 2001.

[2] Fong, J., Wong, H.K., "Replicate relational and XML databases for Internet computing", International Conference on Information Technology: Coding and Computing, 2004.

[3] Funderburk, J. E., Kiernan, G., Shanmugasundaram, J., Shekita, E., Wei, C., "XTABLES: Bridging relational technology and XML". IBM Systems Journal, 2002.

[4] Jacinto, M.H., Librelotto, G.R., Ramalho, J.C., Henriques, P.R., "Bidirectional conversion between XML documents and relational databases", The 7th International Conference on Computer Supported Cooperative Work in Design, 2002.

[5] Jiuchun Gu, Yanguo Qu, Zhenhua Zhang, "Data Sharing and Exchanging Method for Urban Traffic Control System Based on XML", Measuring Technology and Mechatronics Automation 2009.

[6] Jingtao Zhou, Shusheng Zhang, Hongwei Sun, Mingwei Wang, "An XML-based schema translation method for relational data sharing and exchanging", Computer Supported Co-operative Work in Design, 2004.

[7] OuYangZhengZheng, "Application of XML-based Heterogeneous Database Data Exchange Middleware in Ecommerce" IITA Services Science, Management and Engineering, 2009.

[8] Sebastian K Abisch, Daniel Peintner, Jorg Heuer, Harald Kosch, "Efficient and Flexible XML-based Data-Exchange in Microcontroller-based Sensor Actor Networks", IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, 2010.

[9] Tzvetkov, V., Xiong Wang, "DBXML - Connecting XML with Relational Databases", The fifth International Conference on Computer and Informational Technology, 2005.