

Design and Development of a New JAVA Based Conversion Tool from Relational Database to XML Datasets

B. Hemanthkumar* and M.Surendra Prasad Babu**

* Dept., of Information Technology,

RVR & JC College of Engineering, Guntur, A.P. India,

** Dept., of Computer Science & Systems Engg. AU College of Engineering (A),
Andhra University, Visakhapatnam, 530 003, A.P, India,

Abstract: A variety of new applications may be developed using the valuable and tremendous amount of web resources and the web applications that are available in the Internet. To interact with other systems, these web applications needs a standard structured format like XML in developing internet and the web information systems, including web service systems. The response time is an important criterion for Internet-based web services. The time factor plays key role for the end user to survive in severe competition. Getting information from XML datasets is more efficient than from relational datasets with respect to access time and response time. In this paper a new software tool is designed to generate automatically XML datasets from a relational database. An interface to convert Relational data model into XML data model is developed and applied on two test cases: student database and employee database. The results are compared with the existing tools. Finally a Graphical User Interface (GUI) is developed, by using Java, for the generation of an XML report and demonstrated the features of the tool with its advantages.

Key words: XML, SERVLETS, JSP, JDBC.

I. INTRODUCTION:

Web services (WS)[1,2] are basic building blocks, which provide automatic services like on-line transactions, Insurance, Reservation, Trading, and Banking service with high accuracy. Web services are presently described syntactically using XML [3, 4]-based standards, i.e., SOAP [5,6], WSDL[7] and UDDI[8]. To achieve high accuracy and automation, web services are described semantically using some ontology language [9, 10], called semantic web services (SWS)[11]. Quality of service (QoS), is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. Elements of network performance within the scope of QoS often include availability (uptime), bandwidth (throughput), latency (delay), and error rate. It is a critical task to provide and invoke QoS Web services for the success of both service providers and service consumers. Unfortunately, the first generation Web service architectures are not semantic-oriented, and therefore do not provide semantic information about Web services during the publishing, discovery and invocation processes. They

provide solutions based on centralized discovery methods (such

as UDDI) and publish their capabilities and functionalities with a registry and service interface function. These solutions involves high operational and maintenance cost and also do not ensure the required scalability to support a flexible and dynamic environment.

By combining web services technologies, such as SOAP, WSDL and UDDI, with semantic web technologies[12], such as XML, RDF[13] and OWL, a new second generation web , popularly known as 'Semantic Web' can be designed to enable discovery, execution and composition of automated web services with high accuracy. Semantic Web uses XML (Extensible Markup Language) in the place of HTML. An XML file contains not only data but also metadata, semantic information about that data, where as in HTML file contains data only. Therefore XML document is very similar to a database document, and hence it looks like a big database. Semantic Web (SW) aims at making its content not only machine-readable but also machine-understandable. To define machine-readable and machine-understandable content in the semantic web the data is represented in Resource Description Framework (RDF) format [14] which fallows the XML syntax. Here we have developed a new software tool to generate automatically XML datasets from a relational database. It is a GUI developed by using java to convert Relational datasets into XML datasets.

II. BASIC TECHNOLOGIES:

II.A. XML: XML provides a mechanism for structuring a document, so that it can be exchanged and manipulated easily. It is easy to move the XML structured information from one place to another, or from one application to another application. In the real world applications the computer systems and the databases may contain incompatible data formats. However XML data is stored in plain text format; so it is possible to maintain hardware-independent and software independent data. Different applications can share data easily. XML reduces the complexity of exchanging data over incompatible systems. It is well known that XML is the solution to bring structure

to unstructured content. XML makes easy to upgrade new operating system, new applications, or new browsers, without losing data. With XML, the data can be available to all kinds of "reading machines" like handheld computers, voice machines, news feeds, etc. XML documents have data elements, markup declarations (instructions for the XML parser), and processing instructions [15] (for the application program that is processing the data in the document). In XML, one can defines a new nested tag to provide more info about the content of a tag. Attributes of an element can reduce the complexity of the element. Moreover, XML-based data exchange is easy to implement for Web Services.

II.B. SERVLETS: Servlets enable the functionality of the Web Servers [16]. So this is the server side technology to provide a mechanism for developing the server side applications. We can develop platform independent and component based web applications by using Servlets technology. These server side applications can efficiently run on any Servlet enabled Web Server. The Web Server needs Java Virtual Machine to run java servlets. The servlet does not check for the browser compatibility because it runs on the server side. The servlets can access database by using JDBC to develop enterprise applications. The servlets can handle HTTP requests and response in the form of HTTP response. So by using Servlets we can develop portable and secure server side applications. Servlets are most commonly used with the HTTP protocols and are not implemented for specific protocols. To develop Servlets, all the classes are available in the packages `javax.servlet` and `javax.servlet.http`.

II.C. JSP: Java Server Pages is used for developing dynamic web sites. We can write server side scripting in JSP to develop database related web applications. In JSP developers can write java code directly in JSP scriptlets [17], this makes the development process very simple and its maintenance also becomes very easy. The data can be maintained commonly in relational data base; to retrieve data from the relational data base one of the technology used is JDBC in heterogeneous database environment. By using JSP and JDBC it is very easy to develop database related web application. Whenever JSP executed, it generates appropriate servlet for the JSP page, so JSP is treated as a high-level abstraction of java servlets. In JSP developers can define Java code in scriptlet tags and other pre-defined static web markup contents, which is being compiled and executed on the server and send response in the form of an HTML or XML document. All JSP pages are executed within a Java virtual machine (JVM) and use the required Java libraries. These JSP pages are operating system independent and execute on any server which supports JVM. The JSP applications contain JSP scriptlets

as well as the markup elements like html or xml. The Scriptlets contain blocks of java code interpreted by JVM.

II.D. JDBC: The Java Database Connectivity (JDBC) API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases, SQL databases and other tabular data sources, such as spreadsheets or flat files. The JDBC API provides a call-level API for SQL-based database access. JDBC technology allows the use of Java programming language to exploit "Write Once, Run Anywhere" capabilities for applications that require access to enterprise data. With a JDBC technology-enabled driver[22], one can access all the corporate data even in a heterogeneous environment. The main steps involved in JDBC connection are given below

1. Load the JDBC Drivers Using `Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");`
2. Establish Connection Using connection object.
3. Connection
`c=DriverManager.getConnection("jdbc:odbc:dsn","username","password");`
4. Create statement Object as `Statement st=c.createStatement();`
5. Execute sql query as `st.executeQuery(query);`

II.E. DTD: A DTD [19] is made up of three main building blocks: ELEMENT, ATTLIST, and ENTITY. ELEMENT is the main building block of XML documents. In the DTD, XML elements are declared with an ELEMENT. An element definition has the following syntax: `<! ELEMENT element-name (element-content)> element-content` may be EMPTY, or data type, or sequences of children. Because ELEMENT is used to describe elements of a document and each element can contain children elements, the function of these elements is like a class in a structure program. ATTLIST provides extra information about elements. Following is a general syntax of an ATTLIST element: `<! ATTLIST element-name attribute-name attribute type default-value>` `element-name` is the name of element for which we declare an attribute. `Attribute-name` is a name of the attribute we want to declare. `Attribute-type` is a data type and `default-value` specifies default value of the attribute. Finally, ENTITY is used to define a shortcut for a common text in XML. Its syntax is as follows: `<! ENTITY name definition>`. Besides these there are some declarations in DTD, such as CDATA, PCDATA, #REQUIRED, #IMPLIED, etc. Their purpose is to declare the data type or the displaying conditions of elements or attributes in the document.

II.F. RDB: The primary goal of the DBMS [20] is to provide an environment that is both convenient and efficient

to use in retrieving and storing the database information. Relational database system consists of a collection of tables representing both the data and the relationships among those tables. Each table has multiple columns and each column has a unique name. A row in a table represents a relationship among a set of values.

III. PROPOSED SYSTEM ON WEBTIER:

A J2EE application's Webtier [18] makes the application's business logic available on the World Wide Web. The Web tier handles all of a J2EE application's communication with Web clients, invoking business logic and transmitting data in response to incoming requests. A server in the Web tier processes HTTP requests. In a J2EE application, the Web tier usually manages the interaction between Web clients and the application's business logic. The Web tier typically produces HTML or XML content, though the Web tier can generate and serve any content type. While business logic is often implemented as enterprise beans, it may also be implemented entirely within the Web tier. The Web tier typically performs the following functions in a J2EE application: (a) Web-enables business logic, (b) Generates dynamic content. (c) Presents data and collects input, (d) Controls screen flow, (e) Maintains state, (f) Supports multiple and future client types. (g) May implement business logic.

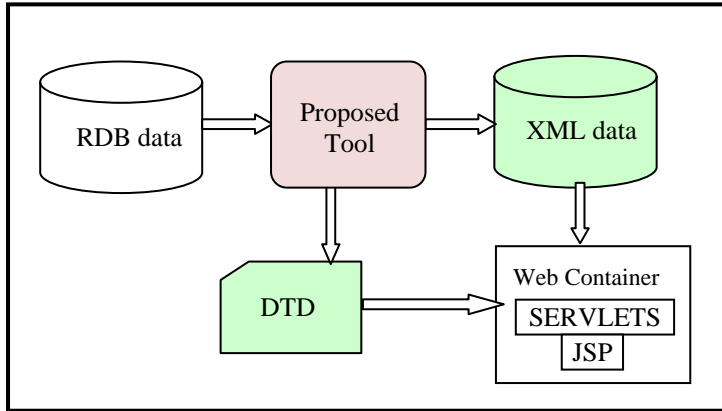


Figure 1: Proposed System on Web tier

III.A. Web Container:

The Web container [18] is a container which is maintained on the web server of the J2EE System. This web container provides the dynamic web services to the end user. The commonly available services are Servlets and Java Server Pages (JSPs) [22]. As shown in figure-1, the proposed tool generates XML datasets as well as the corresponding DTD for the Relational datasets. Web Services like JSPs or Servlets can dynamically access these XML data sets to improve the performance of the system

IV. DESIGN AND DEVELOPMENT OF THE PROPOSED TOOL.

IV.A. Tool Design: Figure: 2 show the designing view of the system which is a GUI that can be developed by using Java to generate XML data sets dynamically. In this system, we have provided different components as shown in the figure 2. The system accepts RDB table as input and generates the XML data and the corresponding DTD

The tool contains the following components:

1. TEXT FIELD to get input,
2. Buttons **GENXMLDATASETS**, **DISPLAYXML**, **DISPLAYDTD**, **DATATABLE**, **TREESTRUCT** and **CLEAR**.

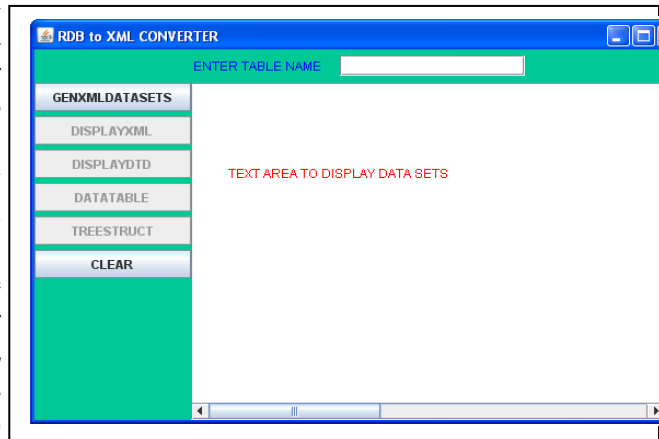


Figure:2 . Design view of the Conversion tool

3. Text Area to display data sets.

Initially the buttons **DISPLAYXML**, **DISPLAYDTD**, **DATATABLE** and **TREESTRUCT** are in disable mode as shown in figure 2. These buttons are enabled only after the generation of XML data sets for the given relational data sets as shown in the figure 3.

IV.B. Tool Description:

The proposed system is based on four phases which are mentioned below:

1. Extraction of the database using SQL language and JDBC
2. Generation of the XML data set.
3. Generate the DTD for the xml data set.
4. Display the data sets in tabular format.

The functionality of the components is as follows:

1. TEXT FIELD: This is input field. The user needs to enter the relational data base table in this field.
2. When the user clicks **GENXMLDATASETS**, the tool takes the database table name from the input field, displays the data and automatically creates an XML data set and corresponding DTD with database table name as xml file name.
3. When the user clicks **DISPLAYXML** the tool takes the XML file name from the input field and displays the corresponding xml data in the text area provided in the interface.

4. When the user clicks **DIPLAYDTD** the tool takes the XML file name from the input field and displays the corresponding DTD in the text area provided in the interface.
5. When the user clicks **DATATABLE** the tool creates a new window as shown in figure 4 and data is in the form of a table. In this window there is an option to sort the data with respect to the given field. There is a text field to enter the field name to sort and the button **SORTDATA** to sort the data as shown in figure 5.
6. When the user clicks **TREESTRUCT** the tool creates a tree structure for the generated XML data sets.
7. When the user clicks **CLEAR** the tool clears the data in the text area provided in the interface as well as it clear the input field.

the Java GUI components. Initially the user needs to give the RDB table name as input in the text field as shown in figure 1. The tool accepts the table name and creates the XML data set. While generating the XML model the table name is treated as the root element for the XML data; each row is treated as a separate record. So the XML file contains a number of records as child elements. In the RDB table each row contains a number of fields (columns). Each field is treated as a child element for the element record. The field values are the values for these child elements. At the same time the tool generates a DTD for all the elements defined in the xml data set. The flow chart as shown in the Figure-6 represents the conversion mechanism of the RDB datasets into the XML datasets.

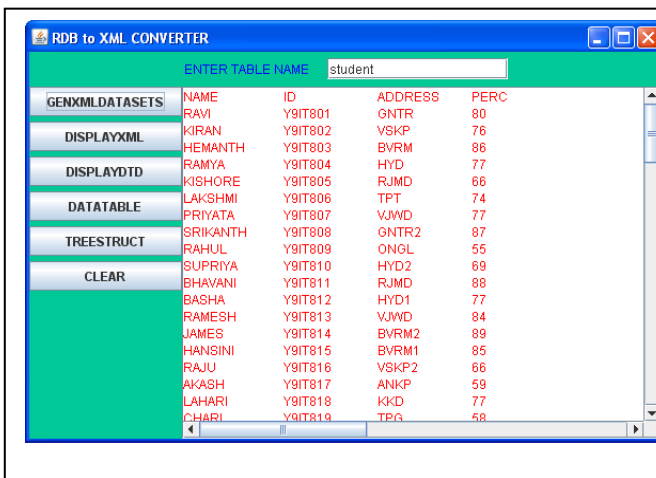


Figure 3: Conversion tool with Active Buttons

V. IMPLEMENTATION METHODOLOGY OF THE PROPOSED TOOL:

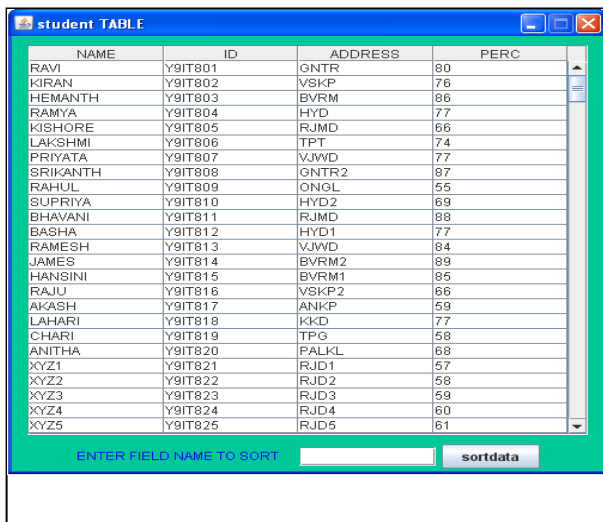


Figure: 4. Data in tabular format

V.A. Methodology.

The flow chart representing the conversion mechanism is as shown in the figure 6. The entire tool is developed by using

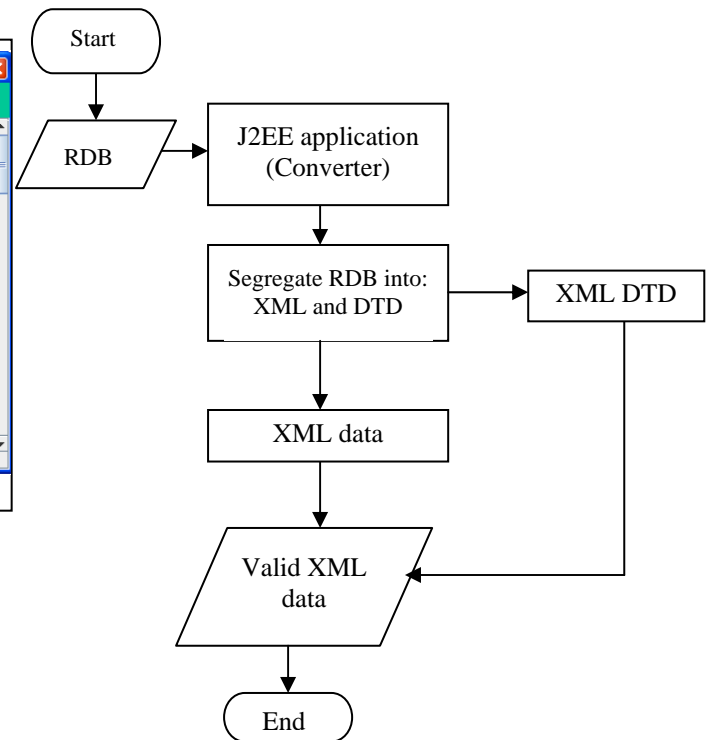


Figure 6: Data flow diagram

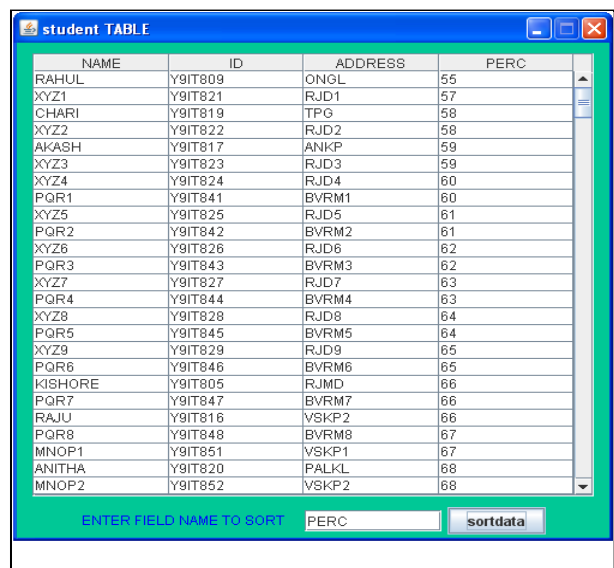


Figure: 5. Sorted data

V.B. Algorithm to generate DTD.

Steps to generate Document Type definition (DTD) for the XML dataset.

- i. Get the table name from the input field and define it as root element for the XML data.
- ii. Define a child element for each record under the root element. Declare more than one record.
- iii. Open the data base connection and get the table data and structure information in to the **Result set Metadata** [22]. Read the field names from this **Result Set metadata** and define these elements as sub elements for the element record.
- iv. Define the data type for these sub elements.
- v. Save the DTD file and then create the XML data file.
- vi. End.

For example the student table structure is as given below:
Student:

STNAME	STID	STADDR
Kiran	Y9it801	Guntur
Ravi	Y9i7802	VSKP
Kumar	Y9it803	HYD

DTD for the above table is

```

<! ELEMENT Student (record+) >
<! ELEMENT record (stname, stid, staddr) >
<! ELEMENT stname (#PCDATA) >
<! ELEMENT stid (#PCDATA) >
<! ELEMENT staddr (#PCDATA) >
    
```

V.C. Algorithm to generate XML data.

Steps to generate the XML data set.

1. Read RDB table from the input field.
2. Establish a connection to access the table.
3. Retrieve the data from the table and maintain it in **ResultSetMetadata**.
4. Create an XML file with root element as <table name>.
5. Creating the element record:
 - a. For each row create an element <record>
 - b. For each record
 - i. get the field name from the **ResultSetMetadata** and create a child element to <record>. The child element name is name of the field i.e. <field>
 - ii. get the value of the field from **ResultSet**. Set this value as the value for the element <field>. Close the element field i.e.</field>

- iii. if another field existing GO to step (i) otherwise close the element record i.e. </record>.
6. If another row existing then GO to Step 5. Otherwise close the root element i.e. </table name>
7. End.

VI. RESULTS.

Dataset 1. Conversion of the student relational database into student XML dataset.

Figure 7.1 shows actual data existing in the student relational database.

Figure 7.2 Shows the DTD for the converted XML dataset of the student database.

Figure 7.3 Shows the XML dataset for the student database after conversion.

Figure 7.4 Shows the Tree structure of the XML data

Figure 7.5 Student data in tabular format.

Figure 7.6 Sorted data on the field student percentage.

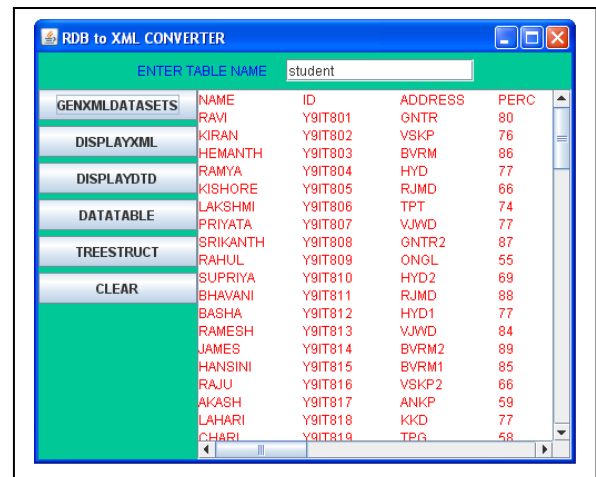


Figure 7.1: Student data

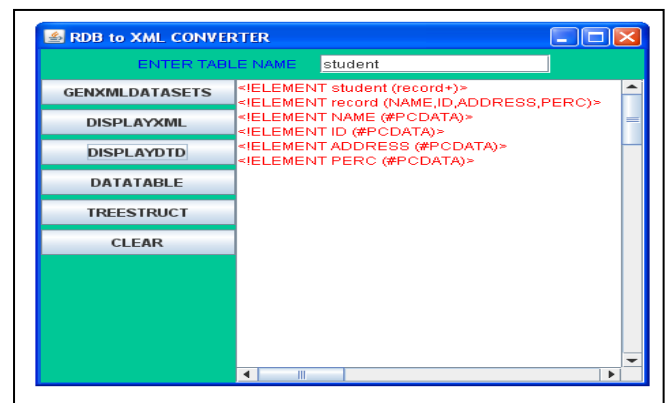


Figure 7.2: DTD for the XML

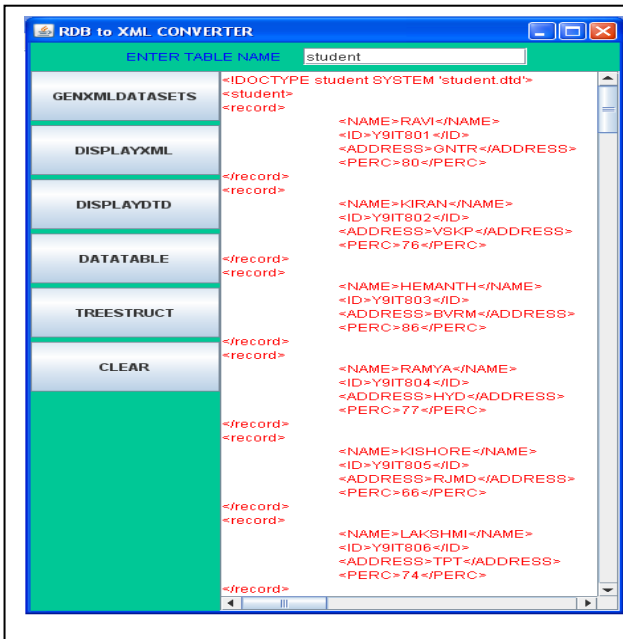


Figure 7.3: student data in XML format

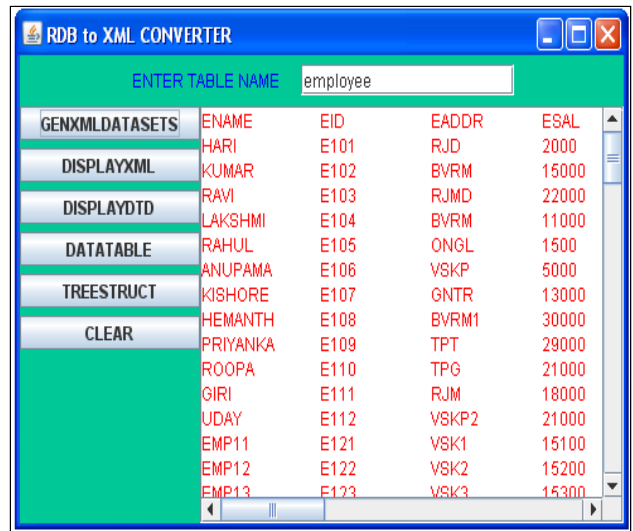


Figure 8.1: Employee Data

NAME	ID	ADDRESS	PERC
RAVI	Y9IT801	GNTR	80
KIRAN	Y9IT802	VSKP	76
HEMANTH	Y9IT803	BVRM	86
RAMYA	Y9IT804	HYD	77
KISHORE	Y9IT805	RJMD	66
LAKSHMI	Y9IT806	TPT	74
PRIYATA	Y9IT807	VJWD	77
SRIKANTH	Y9IT808	GNTR2	87
RAHUL	Y9IT809	ONGL	55
SUPRIYA	Y9IT810	HYD2	89
BHAVANI	Y9IT811	RJMD	88
BASHA	Y9IT812	HYD1	77
RAMESH	Y9IT813	VJWD	84
JAMES	Y9IT814	BVRM2	89
HANSINI	Y9IT815	BVRM1	85
RAJU	Y9IT816	VSKP2	66
AKASH	Y9IT817	ANKP	59
LAHARI	Y9IT818	KKD	77
CHARI	Y9IT819	TPG	58
ANITHA	Y9IT820	PALKL	68
XYZ1	Y9IT821	RJD1	57
XYZ2	Y9IT822	RJD2	58
XYZ3	Y9IT823	RJD3	59
XYZ4	Y9IT824	RJD4	60
XYZ5	Y9IT825	RJD5	61

Figure 7.5: Student data in tabular format.

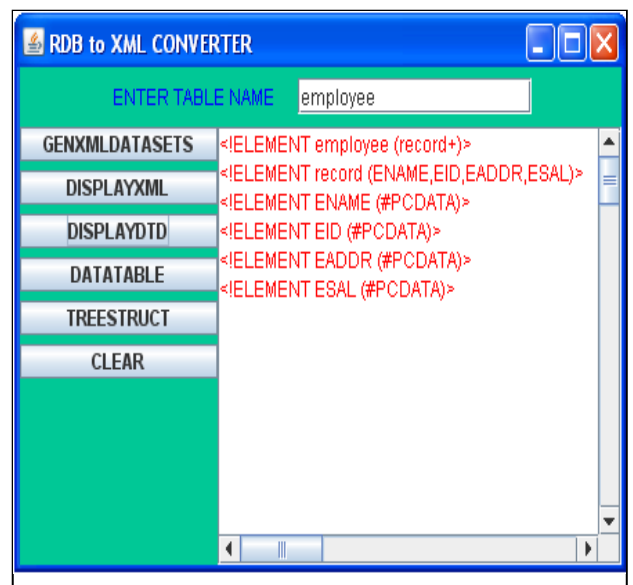


Figure 8.2: DTD for the XML

NAME	ID	ADDRESS	PERC
RAHUL	Y9IT809	ONGL	55
XYZ1	Y9IT821	RJD1	57
CHARI	Y9IT819	TPG	58
XYZ2	Y9IT822	RJD2	58
AKASH	Y9IT817	ANKP	59
XYZ3	Y9IT823	RJD3	59
XYZ4	Y9IT824	RJD4	60
PQR1	Y9IT841	BVRM1	60
XYZ5	Y9IT825	RJD5	61
PQR2	Y9IT842	BVRM2	61
XYZ6	Y9IT826	RJD6	62
PQR3	Y9IT843	BVRM3	62
XYZ7	Y9IT827	RJD7	63
PQR4	Y9IT844	BVRM4	63
XYZ8	Y9IT828	RJD8	64
PQR5	Y9IT845	BVRM5	64
XYZ9	Y9IT829	RJD9	65
PQR6	Y9IT846	BVRM6	65
KISHORE	Y9IT805	RJMD	66
PQR7	Y9IT847	BVRM7	66
RAJU	Y9IT816	VSKP2	66
PQR8	Y9IT848	BVRM8	67
MINOP1	Y9IT851	VSKP1	67
ANITHA	Y9IT820	PALKL	68
MINOP2	Y9IT852	VSKP2	68

Figure 7.6: Student data sorted on

Dataset 2. Conversion of database into employee XML dataset.

Figure 8.1 Shows actual data existing in the employee relational database.

Figure 8.2 Shows the DTD for the converted XML data set of the employee database.

Figure 8.3 Shows the XML dataset for the employee database after conversion.

Figure 8.4 Shows the Tree structure for the employee data.

Figure 8.5 Employee data in tabular format.

Figure 8.6 Sorted data on the field employee salary.

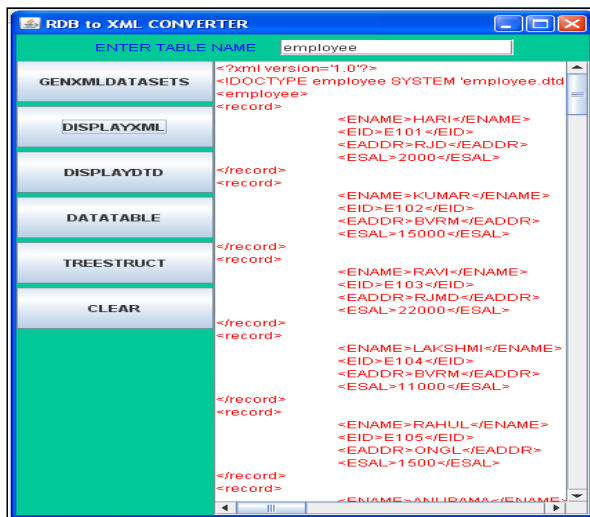


Figure 8.3:Employee data in XML format

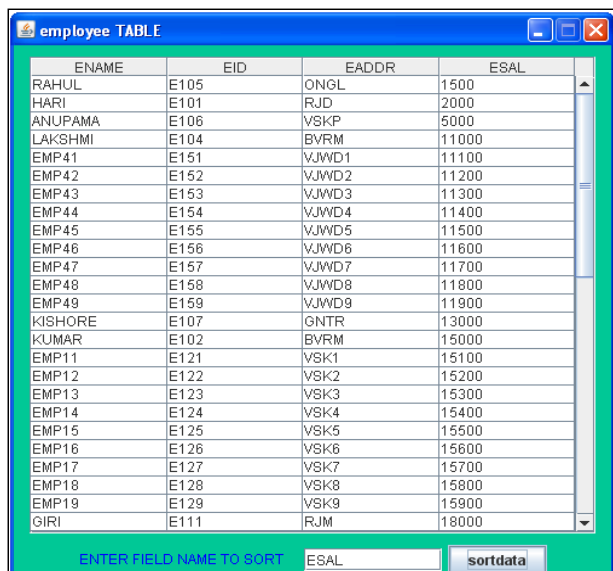


Figure 8.6: Employee data sorted on Salary

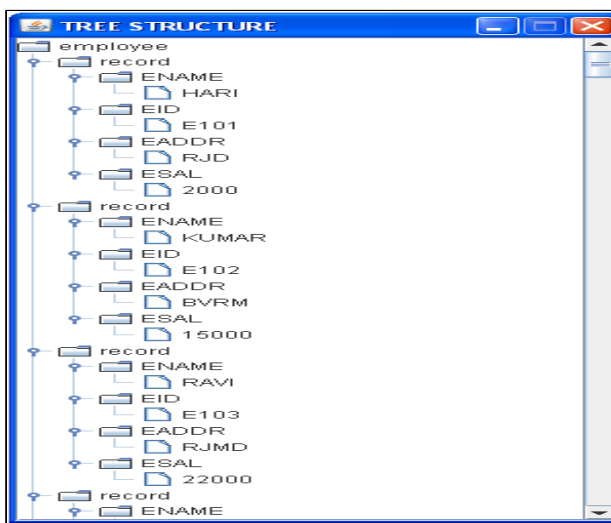


Figure 8.4:Employee data Tree Structure



Figure 8.5: Employee data in tabular format.

VII. COMPARISON WITH EXISTING TOOLS:

Some of the available XML conversion tools are (i). Unidex XML converter2.2 [23], and (ii) RustemSoft XML Converter Standard Edition [24].

(i) **Unidex XML converter2.2:** Developed by Bob Lyons. Bob Lyons is the founder of Unidex which provides B2B integration solutions to its clients in the financial, insurance and telecommunications industries. XML Converter 2.2 is a Java application that performs the conversions: (a) flat file to XML document (b) XML Document to flat file and (c) flat file to flat file. XML Convert uses XFlat schemas to validate the input file and drive the conversion process. An XFlat schema is an XML document that conforms to the XFlat language. XFlat is an XML language for describing flat file formats. XML Convert should run on any platform for which a Java Virtual Machine is available.

Disadvantages:

- The data must be maintained in a flat file
- You must explicitly define schema for this flat file.
- It won't generate DTD or XML schema for the generated XML file.
- The inputs for this file must be a flat file and the corresponding schema file.
- It won't accept any relational data as input.

(ii) **RustemSoft XML Converter Standard Edition:** XML Converter is the Windows based graphical transformation engine for creating XML documents from your database. It allows organizations to quickly and easily share information, captured within corporate data, with anyone in the world by utilizing established XML document definitions. Using this software, enterprises can share data with customers, suppliers or partners as well as distribute data across business units. XML Converter is great for producing fast, static, simple, data-driven XML documents. XML Converter is designed for 32-bit operating systems and works under the Microsoft .NET 2.0 Framework.

Disadvantages:

- It runs only on Microsoft.NET2.0 Framework.
- Because it is developed on Microsoft.NET it runs on Microsoft Windows family.
- It is not portable. i.e. you cannot run this application on any platform.

Advantages of Proposed Tool:

This tool is developed by using java technologies which is a portable language. The proposed system can dynamically generate the xml datasets for the given Relational datasets.

Advantages:

- It can convert any relational datasets into XML datasets.
- Can use this tool on any operating system which supports JVM.
- It generates the XML file and corresponding DTD for the generated XML file.
- It can validate the generated xml file.

VIII. CONCLUSIONS.

The proposed tool takes advantages of the new technologies in the Web like XML, XSL, JSP and SERVLETS. This architecture presents a feasible alternative to produce Web reports in an automatic and easy way. In future, the Semantic Web insures automatic-metadata with a standard XML format that can be interpreted by all the software agents. Web services are a promising technology that will increasingly help to integrate independent systems, provided by business partners participating in multiple enterprises. We can build the new databases based on open standards (XML Web Services) that are accessible, interoperable and promote the use of XML Web Services. In future work we are going to develop a system that converts these XML data sets into RDF which is used to describe Ontologies in Semantic Web architecture.

REFERENCES:

[1] David Booth, Hugo Haas, "Web Services Architecture", Technical report, World Wide Web Consortium (W3C), 2004, <http://www.w3.org/TR/ws-arch>

[2] Ranjit Bose, "Applying Semantic Web Services to Web-based Information Systems and Applications", Proc. of Fifth IEEE International Conference on Information Technology: New Generations, 2008, Las Vegas, Nevada, USA. Pages 1150-1155.

[3] T. Bray, J. Paoli, C.M.Sperberg, McQueen, E. Maler, "XML 1.0 Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation, 6 October 2000, <http://www.w3.org/TR/2000/REC-xml-20001006>

[4] Rafae Bhatti, Elisa Bertino, Arif Ghafoor, James B.D., Joshi, "XML-Based Specification for Web Services Document Security" Journal of Computer archive, Vol.37, No.4, April 2004, pp 41-49, IEEE Computer Society Press, Los Alamitos, CA, USA.

[5] M.Gudgin, M.Hadley, N.Mendelsohn, J-J. Moreau, H. Nielsen, "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, 24 June 2003, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624>

[6] J. A.Miller, D.Palaniswami, A.P.Sheth, K.Kochut, H.Singh. "Web Work: METEOR 2's Web-based Workflow Management System", Journal of Intelligent Information Systems, 10(2) pages 185-215, 1998.

[7] Enk Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarna., "Web Services Description Language (WSDL) 1.1", World Wide Web Consortium note, 20012, www.w3.org/TR/2001/NOTE-wsdl-20010315

[8] Patrick C. K. Hung, Elena Ferrari, Barbara Carminati, "Towards Standardized Web Services Privacy Technologies", Proc. of IEEE International Conference on Web Services (ICWS'04), 2004, pp 174-181, IEEE Computer Society Press, Los Alamitos, CA, USA.

[9] Michael K. Smith, Chris Welty, Deborah L, "Web Ontology Language (OWL) Guide Version 1.0", W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/owl-guide/>

[10] "OWL 2 Web Ontology Language Document Overview", W3C Recommendation, 27 October 2009, <http://www.w3.org/TR/owl2-overview>

[11] Martin Hep, "Semantic Web and Semantic Web Services Father and Son or Indivisible Twins?" IEEE Internet Computing, Vol: 10, No: 2, March-April 2006 pages 85-88.

[12] Tim Berners-Lee, James Hendler, Ora Lassila, "The Semantic Web", Scientific American Digital Magazine, Vol:284, No:5, May 2001, pages 29-37.

[13] Stefan Decker, Prasenjit Matura, Sergey Melnik, "Framework for the Semantic Web: An RDF Tutorial" IEEE INTERNET COMPUTING, IEEE Computer Society, November-December 2000, pages 68-73, <http://computer.org/internet>

[14] Ora Lassila, Ralph R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification" W3C recommendation, Feb. 1999, <http://www.w3.org/TR/1999/REC-Rdf-Syntax-19990222>

[15] Chen Zhifeng, "Research on the Application of XML in Web Services", 2009 IEEE International Conference on Multimedia Information Networking and Security, Wuhan, China, Vol:1, pages 680-683.

[16] Junmin Jia, Chun Meng, Heyue Zhou, "Hierarchical Architecture for Semantic Peer-to-Peer Web Service Discovery", Proc. of 2010 IEEE Int. Conference on Web Information Systems and Mining, pages 166-170.

[17] Mario Guillen R., Victor J. Sosa S., M. Del Rosario Vazquez A., Humberto Hernandez G., "GARP: A Tool for Creating Dynamic Web Reports Using XSL and XML Technologies", Proc. of IEEE Fourth Mexican International Conference on Computer Science (ENC'03), IEEE Computer Society Washington, DC, USA, 2003, pp 54-59, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&number=1232874>

[18] Kun Liu, Shengqun Tang, Jia Wan, Song Ding, Xue Qin, "Improved Layered Architecture for Developing Semantic Web Application", Proc. of 2010 2nd IEEE International Conference on Future Computer and Communication, Vol:3, pp769- 772.

[19] Harvey M. Deitel, Paul J. Deitel, "Internet & World Wide Web How to Program", 4/e, Prentice Hall.

[20] Silberschatz, Henry F. Korth, "Database System Concepts", McGraw-Hill.

[21] Brett McLaughlin, "Enterprise Applications in Java and J2EE", O'Reilly Media, 07-01-2000.

[22] <http://www.oracle.com/technetwork/java/javase/jdbc/index.html>

[23] B2B Integration Solutions from Unidex: <http://www.unidex.com/xflat.htm>

[24] Rustem Soft XML Converter Standard Edition: <http://www.rustemsoft.com/>