

# Interrelation of Data Warehouse and XML Technologies

Goran S. Krstic<sup>1</sup>

**Abstract** - This paper analyses the capability of upgrading of the existent Data Warehouse system by XML technologies with the aim of getting the new quality. Also, this paper presents advantages which have been received by mutual using of these two technologies.

**Keywords:** Data Warehouse, XML, n-tier.

## I. INTRODUCTION

*Datawarehouse* (DW) and XML offer complementary functionality for dealing with data. DW offers efficient storing and retrieving of data, whereas XML offers their easy exchange between applications and passing across network. In order to take advantage of XML's possibilities, it is necessary to convert tables into XML documents. Then, XML tools can be used for further data processing.

Advancement of one DW system by using Oracle XML tools is presented in the paper. Also, advantages obtained by mutual using of DW and XML technologies are presented and analyzed.

## II. DW SYSTEM

DW is today considered to be the most comprising architecture and new information infrastructure which incorporates all concepts and subtypes of information system. The essence of DW philosophy is based on the concept of data integration. The data integration is more important than the sum of data parts. It is important to notice that DW is the structure which is built and not the product which is purchased. *Data Warehousing* represents refinement and aggregation of disparate operative data and provides the basis for the following generations of DSS, ES, OLAP and Data Mining [1].

DW building represents more than pure data coping and allowing the user the tools for decision support. In order to take advantage of DW potentials it is necessary to perform restructuring of data: tables are to be denormalized, new fields for the meeting of user's needs are to be added for sorting, combining and summing of data. New entities are formed and they will enable the data to become real information. New entities occur by merging of entities of origin database or by their derivation. Also, data from external sources as well as trends, prognoses and evaluations can be involved.

<sup>1</sup>Goran S. Krstic, Faculty of Economics, Trg VJ 11, 18000 Nis, Yugoslavia, E-mail: gorca@ni.ac.yu

*Data Warehouse System* (DWS) presented in this paper, has been developed for the needs of executive staff at Faculty of Economics, who have a constant need for following the teaching process. The students database developed within uniform information system of Nis University was used as origin database.

Oracle8i server was installed on local PC computer under Windows 2000 Server and DW database was formed where refined data from origin database necessary for everyday work are loaded (Figure 1).

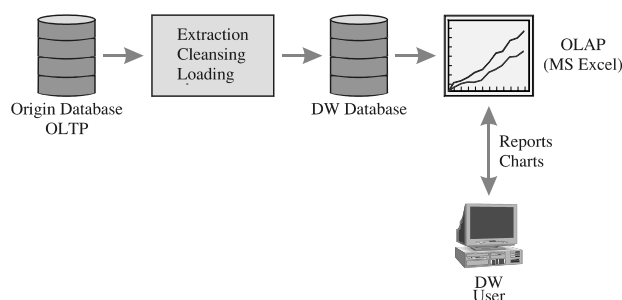


Figure 1. Scheme of DW system realization

The system presented on figure 1 has all advantages offered by DW concept. In addition, by using Excel as OLAP tool, DW user has the possibility of setting ad-hoc query, using the pivot tables, graphic visual presentation, generating of various reports, etc.

Further features of this system derive from classic client-server architecture [2, 3]. A good characteristic of this system is possibility for its upgrading. The client and server side can be upgraded and improved independently. The disadvantage is, that in case of changing of some system parameters, administrating and alterations must be performed on several client computers.

A great deal of useful information is stored in DW system. Web, which offers standard interface, is the ideal choice for presentation of these information to a wide range of users. Natural evolution of this system is addition of the module for generating dynamic web content.

## III. ADVACEMENT OF THE SYSTEM

There are many ways of generating dynamic web content, such as CGI (*Common Gateway Interface*), JSP (*JavaServer Pages*), ASP (*Active Server Pages*), OAS (*Oracle Application Server*), etc. OAS should be especially emphasized for a close integration of program environment offered by PL/SQL and HTML pages, as well as for the robustness and scalability. In fact, OAS itself can be performed on several computers, whereby n-tier architecture is accomplished in full sense.

Although these approaches have their own advantages, using of XML for creating of dynamic pages, regarding that the data are separated from their presentation has a great deal of advantages:

- It is possible to present one and the same document in many different ways by applying various XSL stylesheets, without additional programming.
- It is possible to change data presentation easily, editing only XSL stylesheets, which is in its base textual file.
- An appropriate stylesheet can be applied selectively for transformation of XLM documents into appropriate format adjusted to the device that is used.
- By the development of the advanced (object, object-relational, temporal, space and scientific) databases and multimedia, the data occurring on Web become more and more complex. HTML is not suitable for the description of these types of data. Solutions for dynamic generating of web content, such as JSP, ASP, and other, are using complex scripts and programs together with HTML pages for combining of these data with web pages which represents “the way round”. XML is better solution, because by its concept of extensible it offers enough space for the involvement of these complex structure data.

By merging of the advantages that offer the using of XML and servlet, Oracle has developed technology XSQL Pages, the nucleus of which constitutes XSQL Servlet tool. The system with additional module for generating of dynamic web content is presented on Figure 2.

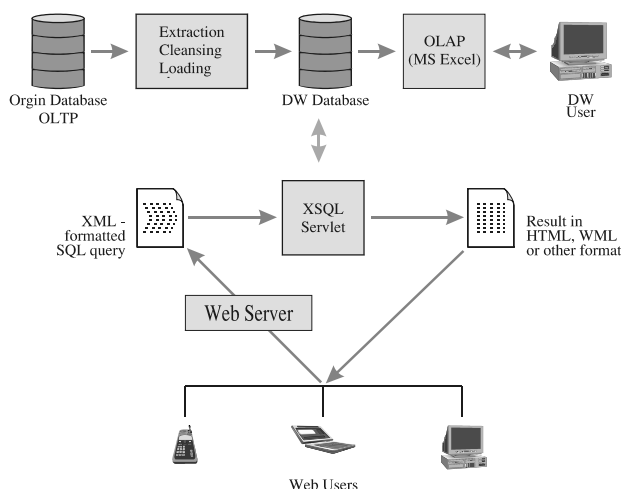


Figure 2. The advanced system.

Now, the system in the branch toward web user gets 3-tier architecture:

- on the client side there is the so called thin client consisting of web browsers,
- middle layer constitutes XSQL Servlet and applications are performed on it. The input and output of application are performed through client and by means of web browser.
- the third layer remains the same as in the client-server architecture and represents the database server.

The program logic is now in the middle layer. Its modification, upgrading and testing are performed in a centralized site. The client part of application is brought down to the web browser, whereby simplicity and uniformity of very applications are achieved. The time necessary to learn and adjust to a new (modified) application is much shorter.

The XSQL Servlet is tool that processes SQL queries and outputs the result set as XML. It consists of the following components: XML Parser, XSQL Page Processor, XML SQL Utility and XSLT Processor.

The detailed representation of these tools can be found in [4]. In Figure 3 the flow of data, from the user to Servlet and back is presented.

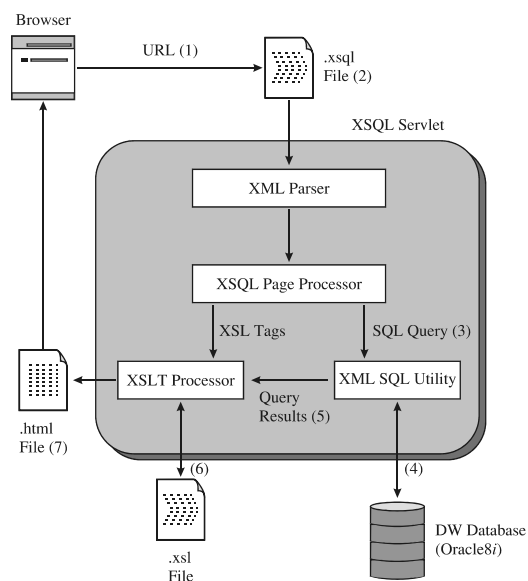


Figure 3. The flow of data from user's request to receiving the results

The sequence of events is as follows:

1. The browser sends request to the servlet compatible web server who passes it to XSQL Servlet. URL contains the name of the target XSQL file (.xsql) which contains XML formatted SQL query and optional parameters.
2. The Servlet passes XSQL file to XML parser, which performs parsing and creates API for accessing the XML content.
3. XSQL Page Processor uses API to pass SQL commands found between <query> and </query> tags to the XML SQL Utility. It also passes XSL commands to the XSLT processor.
4. XML SQL Utility (XSU) is a tool which converts data from database tables into XML documents (SQL-to-XML mapping). XSU sends SQL query to the underlying Oracle8i database, which returns result.
5. XSU sends the result to the XSLT processor as XML formatted text. The result is embedded in the XML file, in the same location as the original <query> tags.
6. The XSLT Processor can transform the result into any format defined by the XSLT stylesheet (.xsl file).

7. The XSLT Processor passes the completed document back to the browser for the presentation to the user.

#### IV. THE DEVELOPMENT OF APPLICATIONS

The XML solutions operating nowadays are mostly connected with some of the existent software products. Various software producers use XML as a language for data description. They have created tools which convert information from database into XML documents and present them in the browsers. Thereat, each product has its own specific traits. This paper presents the development of applications by using Oracle tools. An effort has been made to emphasize as it is possible the conceptual advantages of the XML access, whereas the specific traits of tools are minimized.

##### A. Searching and generating dynamic web content

As an example of the development of application, DW table ANALYSIS(year, term, exam, total, pass, fail, percent) was used, which contains summed data on the efficiency in taking exams according to exam, term, and schoolyear. The following simple XSQL document *analysis.xsql* performs the query on this table:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="rowcol1.xsl"?>
<page connection="student" xmlns:xsql="urn:oracle-xsql">
<xsql:query null-indicator="yes">
```

```
SELECT exam, total, pass, fail, percent
FROM analysis
WHERE year ='{@year}'
AND term ='{@term}'
```

```
</xsql:query>
</page>
```

The query is designed in the way that there exists possibility of setting the parameters, *year* and *term*, by means of which searching conditions are specified. XSQL document also specifies that XSL stylesheet *rowcol1.xsl* will be applied to the resulting XML document that was impossible to present here due to its largeness.

XSL stylesheet represents template by which XML document will be presented to the user. It defines the structure of the document and determines the location in the resulting document where data will be embedded. More details on stylesheet can be found in [4, 5]. The next URL:

<http://160.99.71.3/xsql/analysis/analysis.xsql?year=2001/2002&term=January>

calls XSQL document *analysis.xsql*, which returns the results of taking exams in examination period January 2001/2002. The result is presented on Figure 4.



EXAM	TOTAL	PASS	FAIL	PERCENT
Building	28	28	22	42.07
Business Analysis	8	5	3	42.5
Business Decision	175	45	130	25.71
Business Geography	155	48	107	31.61
Business Policy	8	5	3	62.5
Business of Agriculture	84	48	36	57.14
Business of Business I	133	38	95	28.57
Business of Business II	143	28	115	19.58
Business of PS2	91	33	58	36.26
Business of Industry	72	33	39	45.83
Business of Trade	101	33	68	32.67
English Language I	29	18	11	62.07

Figure 4. The presentation of result by using *rowcol1.xsl*

##### B. Different data presentations

One of the goals of XSL designing was to enable transforming of documents from one format into the other, supporting various user's needs. Thus, for one and the same query, i.e. for the same *analysis.xsql* document we can get different data presentation altering only the stylesheet. Instead of stylesheet *rowcol1.xsl*, by applying of stylesheets *rowcol2.xsl*, the result presented in Figure 5 is obtained.



EXAM	TOTAL	PASS	FAIL	PERCENT
Building	28	28	22	42.07
Business Analysis	8	5	3	42.5
Business Decision	175	45	130	25.71
Business Geography	155	48	107	31.61
Business Policy	8	5	3	62.5
Business of Agriculture	84	48	36	57.14
Business of Business I	133	38	95	28.57
Business of Business II	143	28	115	19.58
Business of PS2	91	33	58	36.26
Business of Industry	72	33	39	45.83
Business of Trade	101	33	68	32.67
English Language I	29	18	11	62.07
English Language II	40	25	15	62.5

Figure 4. The presentation of result by using *rowcol2.xsl*

##### C. Two levels of data selectioning

XSL in itself includes query language XPath, which gives it a great deal of possibilities for searching XML documents, similar to those in SQL searching databases. Thanks to this fact it is possible to reach two levels of data selectioning:

1. By using SQL commands desirable data are selected and transformed into XML document,
2. After that, by using XSL, from XML document particular data can be selected and presented differently representing the information of a higher level in a given context.

For example, for the analysis of efficiency in taking exams, it is necessary to emphasize those exams having the passing percentage lower than 30%. The template and the result (Figure 6) are as follows:

```

<xsl:template match="ROW">
  <xsl:choose>
    <!--If the passing is lower than 30%
    the subject is presented by red colour -->
    <xsl:when test="PERCENT &lt; 30">
      <tr style='color:#ff0000'>
        <xsl:for-each select="*">
          <td>
            <xsl:apply-templates select='.'/>
          </td>
        </xsl:for-each>
      </tr>
    </xsl:when>
    <xsl:otherwise>
      <tr>
        <xsl:for-each select="*">
          <td>
            <xsl:apply-templates select='.'/>
          </td>
        </xsl:for-each>
      </tr>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

```

Exam	TOTAL	PASS	FAIL	PERCENT
Building	28	26	22	92.9
Business Analysis	8	5	3	62.5
Economics Database	115	40	180	34.71
Economics Geography	105	48	59	45.71
Economics Politics	4	0	1	0.00
Economics of Agriculture	84	40	39	47.62
Economics of Botany I	133	59	98	44.36
Economics of Botany II	163	26	117	15.95
Economics of PS2	91	31	40	34.07
Economics of Industry	73	39	39	52.05
Economics of Trade	201	83	70	41.29
English Language I	39	18	43	41.03

Figure 6. Additional selectioning of data within XML document

This was possible to achieve thanks to the conceptual advantage of XML over HTML, i.e. in XML we can give name to each part of the data, whereas in HTML we had only anonymous blocks of data which are put between standard HTML tags. It is also said that XML is “aware” of the data.

#### D. The control of presentation

Thanks to the separation of data from their presentation, XML enables an easy control of the resulting documents presentation. In the .xsql document heading several `<?xml-stylesheet?>` processing instructions can be found and optional attribute *media*, on the basis of which XSQL Page Processor tries to apply an appropriate stylesheet that corresponds to a current User-Agent value (string in HTTP heading). XSQL Page processor compares the attribute media

value with the User-Agent value received through HTTP request. In case there is no matching, the default stylesheet is applied. For example, the following .xsql document ...

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" media="Palm"
  href="analysisPP.xsl"?>
<?xml-stylesheet type="text/xsl" media="EricssonR320/R1A"
  href="analysisWML.xsl"?>
<?xml-stylesheet type="text/xsl"
  href="analysis.xsl"?>
<xsql:query connection="student" xmlns:xsql="urn:oracle-
xsql">
  SELECT * FROM analysis
</xsql:query>

```

will use *analysisPP.xsl* for PDA Palm Pilot, *analysisWML.xsl* for mobile phone and *analysis.xsl* in all other cases.

## V. CONCLUSION

By mutual using of DW and XML technologies a new quality in creating web applications is obtained.

DW enables the user to get the information from a great deal of prepared, clarified and aggregated data. Thanks to its concept of separating the data from the presentation, XML enables different presentation of data adjusted to the user's demands and devices.

Further advancements of the realized system involve the development of applications for the support of micro browsers on different WAP devices such as personal digital assistants and mobile phones.

## REFERENCES

- [1] Nedo Balaban, Živan Ristić, *Sistemi podrške odlučivanja*, Ekonomski fakultet Subotica, Subotica, 1998.
- [2] Petrou, C.; Hadjiefthymiades, S.; Martakos, D., “An XML-based, 3-tier scheme for integrating heterogeneous information sources to the WWW”, proc. Tenth International Workshop on Database and Expert Systems Applications, 1999.
- [3] Volčanšek B., Stanković M., “Prikaz Oracle-ove mrežne arhitekture i njeno koršćenje u razvoju aplikacije za OnLine prijavu radova”, YU INFO 2000, Simpozijum o računarskim naukama i informacionim tehnologijama, 2000.
- [4] Brad Wait, “Using XML in Oracle Database Applications”, June 2001, ([http://technet.oracle.com/tech/xml/info/htdocs/otnwp/xml\\_custom\\_presentation.htm](http://technet.oracle.com/tech/xml/info/htdocs/otnwp/xml_custom_presentation.htm)).
- [5] XSL Transformations (XSLT), (<http://www.w3.org/TR/xslt.html>).