

XML TECHNOLOGIES FOR ELECTRONIC COMMERCE

Lecturer **Rocsana BUCEA-MANEA-ȚONIȘ**, Lecturer **Radu BUCEA-MANEA-ȚONIȘ**
„Spiru Haret“ University

REZUMAT. Standardizarea informației electronice în vederea facilitării globalizării proceselor de afaceri a devenit o cerință imperativă pe piața mondială a soluțiilor e-business. Avantajele tehnologiilor de tip XML sunt maximizate de serviciile web și ebXML - în cazul proceselor de afaceri - care asigură un mecanism rapid de procesare pentru accesul direct, fără mijlocirea sistemului de gestiune a bazei de date. Articolul demonstrează utilitatea acestor tehnologii prin intermediul studiului de caz.

Cuvinte cheie: XSL, XSLT, XML, ebXML, CPA.

ABSTRACT. Standardization of electronic information in order to facilitate the globalization of business processes has become imperative on the world market of e-business solutions. The benefits of XML-based technologies are maximized by ebXML for business processes and web services which provides a mechanism for fast processing for direct access, without the mediation of the database management system. The article demonstrates the usefulness of these technologies through case study.

Keywords: XSL, XSLT, XML, ebXML, CPA.

There are currently two technologies for ensuring the operation of electronic commerce: ebXML and web services. ebXML (Electronic Business using eXtensible Markup Language: XML e-business) is an initiative sponsored by OASIS and UN/CEFACT in order to provide an open infrastructure based on a unique set of concepts of XML architecture enabling the globalized use of broad-based electronic business information, in an interoperable, secure and standardized way for all users. This architecture enables a better implementation of the e-business solutions [1].

EbXML standard describes both the basic source of information used in the business messages and the business processes (such as offering an expert advice, the purchase of goods for resale, offering a professional service). These processes involve the exchange of information between two or more business partners according to a certain logical and predictable order, so that they can be understood by the other organization [2].

ebXML provides a framework for defining the business processes, business documents and electronic exchange of business documents. Business Process Specification Schema (BPSS) provides the definition of an XML document that describes the way in which the organization shall conduct its business, specifying partners,

roles and collaboration rules. Finally provides a document attesting the business process. Business processes allow defining the business collaboration, such as business transactions that are based on a document flow. Transactions succeed after a certain logical planning. This is specified using the ebXML standard chart of activities [3].

To facilitate the exchange of messages in XML format between companies in the international business environment, it emerged the need of creating terms repositories used in the standard business vocabulary. In addition, for easy identification of business partners there have been created firms directories with metadata about objects themselves and the mappings of the relationships established between them.

While web services covers all types of activity on the Internet, ebXML deals only with the business field, allowing the sharing of well-defined business documents in a safe manner, establishing a cooperation protocol between the business partners - Collaboration Protocol Agreement (CPA). From the technical point of view, web services require co-existence of service provider and many customers for the services provided, while ebXML facilitates the establishment of direct business relations with potential partners [4].

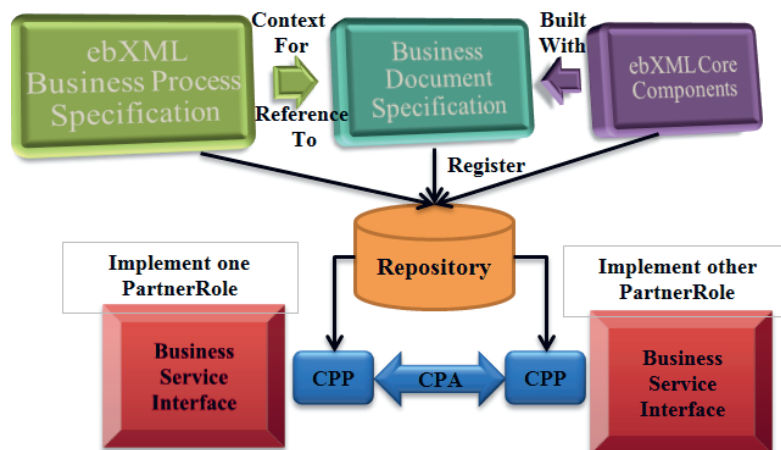


Fig. 1. BPSS: Business Process Specification Schema [3].

CASE STUDY

Online databases require a rapid processing mechanism for direct access, without the mediation of database management system. XSL transformations provide a mechanism, both on client and server side, which eludes the SQL requests to database Server. Thus, the access time to data is reduced and the connection overloading between client and server is avoided. Furthermore, programming languages designed for web include by default parsing modules of XML documents, such as SimpleXML, DOM for PHP and SAX for JAVA, respectively. Standard programming languages may access libxml++ library for

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="s.xsl"?>
<angajati>
<angajat nr="1" nume="Ionescu Pop" vechime="5" departament="Contabilitate"/>
<angajat nr="2" nume="Vasile Radu" vechime="3" departament="Montaj"/>
<angajat nr="3" nume="Stamate Camelia" vechime="10" departament="Secretariat"/>
<angajat nr="4" nume="Vintila George" vechime="1" departament="Montaj"/>
<angajat nr="5" nume="Costache Silvia" vechime="7" departament="Secretariat"/>
<angajat nr="6" nume="Secareanu Andrei" vechime="0" departament="Montaj"/>
</angajati>
```

The data is returned by specifying the absolute path, including the root tag to the node that will contain the data referenced in the template, for example:

```
<xsl:value-of select="xpnetdiag/title" /> ,
```

returns the contents of the tag title, whose parent is the root node xpnetdiag.

To return iteratively the same type tags and located in the same level from the root node, we recommend the instruction for-each, specifying the associated leaves nodes tag near select property, for example:

```
<xsl:for-each
select="xpnetdiag/component"> ,
```

Linux programming environments and Xerces for C from Apache XML Project initiative [5]. Other languages that provide XML transformations are XQuery and LINQ – beginning with 3.5 version of .Net Framework platform from Microsoft.

In the .xsl template file the HTML content is marked with the referenced XML data islands by using XSL tags from an external XML file. The XML data source shown below contains the organizational scheme of the firm. XSL transformation allows the grouping of employees on departmental structures and descending ordering of employees after their seniority in the Department.

enables the return the content of all component tags, that are in parent-child relationship with the root node xpnetdiag.

For sorting the list obtained before after a specified value in the component tag, we recommend the sort directive, specifying the label name with the preceding symbol @ and the sort order, near select and order properties, for example:

```
<xsl:sort select="@startDiagnosisTime"
order="descending" /> ,
```

enables descending ordering of values contained in the component tags after the values associated to the label startDiagnosisTime.

Displaying only the value of a node can be achieved by using the directive when followed by the logic test, near test property, for example:

```
<xsl:when test="@status='confirm'">
<xsl:value-of select="/confirmText" />
<xsl:value-of select="@name" />
</xsl:when>
```

allows displaying the value obtained by relative path of confirmText tag and the name property value, only when the value of the status label is confirmed. Some may observe that when directive is associated with the terminal tag

</xsl:when>, as in structured programming languages.

Grouping by a criterion is achieved using for-each-group directive after the department key has been generated, to go through all employees, having as value the property @department. Every grouping interval will be parsed in order to display the values associated with child nodes nr, nume and vechime, as shown in the following figure 2 (6).

The result of XSL transformation is printed in the figure number 3.

```
1 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
2
3   <xsl:key name="departament" match="angajat" use="@departament" />
4
5   <xsl:template match="angajati">
6     <html><head><title>Grupare folosind XSL</title></head>
7     <body>
8       <TABLE BORDER="1"><TR><TD>DEPARTAMENT</TD><TD>ANGAJAT</TD></TR>
9       <totiangajatii>
10        <xsl:apply-templates select="angajat[generate-id(.)=generate-id(key('departament',@departament)[1])]" />
11      </totiangajatii>
12    </TABLE>
13  </body></html>
14 </xsl:template>
15
16 <xsl:template match="angajat">
17   <TR><TD><xsl:value-of select="@departament" /></TD>
18   <TD>
19     <departament value="@departament">
20     <TABLE BORDER="1"><TR><TD>NR</TD><TD>NUME</TD><TD>VECHIME</TD></TR>
21     <xsl:for-each select="key('departament', @departament)">
22       <TR>
23         <xsl:if test="(position() mod 2 = 1)">
24           <xsl:attribute name="bgcolor" value="#E0E0E0" />
25         </xsl:if>
26         <TD><xsl:value-of select="@nr" /></TD>
27         <TD><xsl:value-of select="@nume" /></TD>
28         <TD><xsl:value-of select="@vechime" /></TD>
29       </TR>
30     </xsl:for-each>
31   </TABLE>
32 </departament>
33 </TD></TR>
34 </xsl:template>
35 </xsl:stylesheet>
```

Fig. 2. The s.xsl file printed.

DEPARTAMENT	ANGAJAT		
Contabilitate	NR	NUME	VECHIME
	1	Ionescu Pop	5
Montaj	NR	NUME	VECHIME
	2	Vasile Radu	3
	4	Vintila George	1
	6	Secareanu Andrei	0
Secretariat	NR	NUME	VECHIME
	3	Stamate Camelia	10
	5	Costache Silvia	7

Fig. 3. The result of viewing s.xml data source file with Internet Explorer 9.

CONCLUSIONS

Databases and data warehouses can no longer meet the full requirements of modeling and implementing business processes which take place today in real time. XML technologies enable aggregation and query of data sources using buffer memory of modern browsers, without imposing an additional consumption of hardware and energy resources. This is shown in the case study presented in the article.

BIBLIOGRAPHY

- [1] Wikipedia: <http://en.wikipedia.org/wiki/EbXML>
- [2] ebXML Tutorial: http://www.tutorialspoint.com/ebxml/ebxml_business_process.htm
- [3] ebXML Specifications: <http://www.ebxml.org/specs/ebBPSS.pdf>
- [4] [http://classes.soe.ucsc.edu/ism211/Winter09/ebxml-
ws.pdf](http://classes.soe.ucsc.edu/ism211/Winter09/ebxml-ws.pdf)
- [5] Apache XML Project: [http://xerces.apache.org/xerces-
c/](http://xerces.apache.org/xerces-c/)
- [6] [http://stackoverflow.com/questions/2146648/how-to-
apply-group-by-on-xslt-elements](http://stackoverflow.com/questions/2146648/how-to-apply-group-by-on-xslt-elements)