

Multilingual Web Application Development on Unicode Database using Oracle XDK 10g

Daniela C. Nikolic¹, Jovanka D. Cekic², Milentije A. Maksimovic³

Abstract – Data incorporated in XML are more self-describing and portable and can be easily shared, transformed, and transported across applications and platforms. In this paper we present a method for developing multilingual Web applications using Oracle XDK 10g - a standards-based set of components, tools and utilities – in order to provide automatic content delivering and update from Oracle10g database backend.

Keywords – multilingual web application, Oracle, Java, XML, XDK.

I. INTRODUCTION

Development of the Internet has provided the companies around the world with the opportunity to optimize and expand their businesses. More than 90% of the world population is non-native English speaking. The challenge and the barrier towards a success on the global market is the development of applications with the support for different language and cultural requirements. A truly global Internet application needs to support data exchange in a wide range of character sets, and the user interface must present data in a format that matches the local convention of the user's language and country.

The Oracle 10g Database and Application server represent a fully internationalized platform for developing and deploying Internet applications with multilingual support. Oracle provides full support for Unicode 3.2 standard for storing, retrieving and updating data in different languages.

There are several approaches in the development of global Internet applications on Oracle 10g platform. [4] The solution presented in this paper is based on using one centralized database with a Unicode character set, and one instance of Oracle Application server. The application running on Oracle 10g AS is configured to support several locales simultaneously, so that each client can use different character set.

Oracle Application Server provides a possibility of developing applications using different programming languages and corresponding Web development environments. The HR application presented in this paper was

developed using Java and XML technologies with XSLT extension function facility provided by Oracle XDK 10g and Oracle Application Server Containers for J2EE (OC4J).

II. XML AND DATABASE

XML is a markup language which can be run on any platform, operating system or environment and it is made in such a way to give designers mechanisms for better description of their content. XML does this by letting designers write their own document-type definitions – DTDs – which describe tags and attributes sets, and can be used for describing specific kinds of content. [1]

XML is fully internationalized for both European and Asian languages, with all conforming processors required to support the Unicode character set in both UTF-8 and UTF-16 encodings.

XML provides context for words, specifies document structure, and allows effective search capabilities. It is flexible, accessible and provides a universal format. [2]

Databases and XML offer complementary functionality for data storing. Database offers efficient retrieval, database administration, business intelligence, recovery tools and procedures, and blocking operations that compromise data integrity. XML offers easy information exchange, and interoperability between applications.

III. ORACLE XML SUPPORT

To help application developers in taking advantage of XML, Oracle Database 10g includes Oracle XML Developer's Kit (XDK). Oracle XDK 10g is a set of components, tools and utilities in Java, C, and C++ and available in Oracle Database 10g and Oracle Application Server 10g that ease the task of building and deploying XML-enabled applications.

Components of Oracle XDK are: XML Parsers, XSLT Processors, XSLT VM, XML Schema Processors, XML Java Beans, XML Class Generator, XML SQL Utility, XSQL Servlet, XML Pipeline Processor, and TransX Utility.

The overall XML infrastructure in Oracle database includes:

1. XMLType tables and views which provide the storage of XML data.
2. XML DB Repository which provides an XML document repository that is optimized for handling XML documents.
3. PL/SQL and SQL/XML functions which allow XML operations on SQL data and XML content.
4. Java and C XML programming which can be used to add functionality by building external procedures, such as creating

¹Daniela C. Nikolic is with the University of Nis, Univerzitetski trg 2, 18000 Nis, Serbia and Montenegro, E-mail: eli@ni.ac.yu

²Jovanka D. Cekic is with the University of Nis, Univerzitetski trg 2, 18000 Nis, Serbia and Montenegro, E-mail: jovanka@ni.ac.yu

³Milentije A. Maksimovic is with the University of Nis, Univerzitetski trg 2, 18000 Nis, Serbia and Montenegro E-mail: maxa@ni.ac.yu

a Java Stored Procedure for SAX stream-based XML processing. [3]

The Oracle XDK provides the functionality to use standards-based interfaces to access, modify, transform, and validate XML documents.

IV. DESIGNING XML DATABASE FOR WEB APPLICATIONS

Before you start to design XML application using Oracle XML DB, you have to choose how to store XML data in a database, how to retrieve or generate XML, and how to create proper indexes for searching the content in the XML documents.

1. Storing XML data - There are different ways to store XML documents inside an Oracle XML database, and each of them offers different trade-offs in both performance and functionality. You can store XML documents in CLOB XMLTypes, XML Schema-based XMLTypes, relational tables, or external tables. In application presented in this paper CLOB XMLTypes are used, because this kind of storage best preserves the original format of XML documents and gives the maximum flexibility for XML schema evolution. Storing an intact XML document in a XML Type CLOB is a good strategy if the XML document contains static content that will only be retrieved as a whole or updated by replacing the entire document.

2. Loading XML data - Depending on the XML storage option chosen, you can use one of the associated data loading strategy in Oracle DB 10g: SQL*Loader, XML SQL Utility, TransX Utility, XSQL Servlet, HTTP / Web DAV, or FTP Interfaces. Upon insertion, XML documents are checked and only committed if well formed.

3. Generating and retrieving XML data - Oracle Database 10g provides the built-in support for XML, which greatly simplifies the retrieval of XML content and the conversion between relational datasets and XML-formatted data. You can generate XML from SQL data either with SQL XML functions or with PL/SQL package DBMS_XMLGEN.

4. Searching XML data - By storing XML data in Oracle DB 10g, you can create SQL indexes and enable users to efficiently search XML data. Oracle DB 10g provides two types of search on XML: the Xpath-based search provided along with the XML Type functions and the full text search using the Oracle DB component, Oracle Text. Oracle Text can provide both content and path indexes to search, but data retrievals need to be done by processing the whole document. [3]

V. SYSTEM ARCHITECTURE

XML is widely used in content management and Web publishing systems. One of the reasons is that data in XML format can easily be transformed by XSLT to various presentation formats, such as HTML, WML (Wireless Markup Language), SVG, or any other Web publishing format that clients request.

Application module presented in this paper represents human resource application that supports English and Serbian language, with XML data exchange.

Fig. 1 represents architecture of an Oracle 10g AS multilingual application and Oracle solution for XML document storage. It is mainly written in Java, using Java beans, and Java Server Pages (JSPs). It uses the Unicode capabilities available in Java, XML, JDBC, and the Oracle10g XML DB to support multilingual data and a multilingual user interface.

Clients access the Web application through a Web browser, passing chosen language as a parameter, which is then stored in the session data. The request is being redirected to the JSP page that instantiates a Java bean with business logic. The Bean accesses the database and extracts the XML data. Those data are transformed by XSLT JSP tag into the HTML UTF-8 encoded and than sent back to the client. An example is shown on the Fig. 2.

Database stores XML documents in XMLType column along with the appropriate encoding.

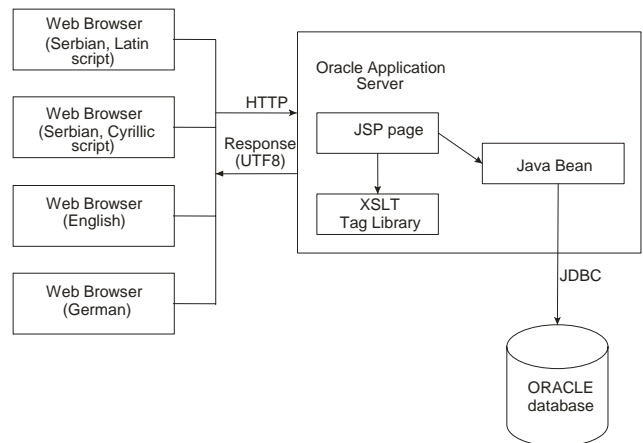


Fig. 1. System Architecture

VI. MULTILINGUAL SUPPORT

All the programming languages and technologies used for developing of this Web application have Unicode support.



Fig. 2. Search page for HR application

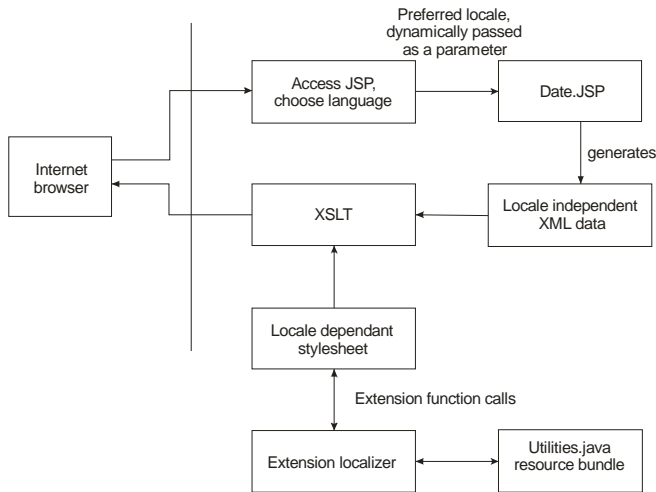


Fig. 3. Diagram of the data localization process

After the user selects a language, it is stored in the session parameters and used in data localization process, shown in Fig. 3. JSP first extracts locale-independent XML data from the database, and then forwards them to the XSLT transformation module. This module then transforms the data using locale-dependent XSL stylesheet (which also uses locale-dependent resource bundle with localized strings).

Multilingual support for the application is provided in the following way:

1. The localizer classes are used in this application to retrieve Locale-specific user interface prompts/resources. Every method accepts a valid ISO language code and returns the user interface prompt/resource string in that language. This application supports two Locales: English (en) and Serbian (sr). XSLT Stylesheet calls java function `getInputProperty`:

```

public static String getInputProperty(String s, String msg)
{
    Locale l = new Locale(s);
    String lang = l.getLanguage();

    if (lang.equalsIgnoreCase("en") ||
    lang.equalsIgnoreCase("sr")) {

        // determine the session locale
        Locale locale = new Locale(lang, "");
        // get the resource bundle for the locale
        ResourceBundle bundle =
        ResourceBundle.getBundle("ApplicationMessages",
        locale);

        // return the translated message
        return bundle.getString(msg);
    }
    return "Error in Localizer";
}
  
```

2. Depending on the language function, corresponding file with all the prompts, messages and resources in that language is called. Example of English file:

```

bannertitle1=HUMAN RESOURCE
bannertitle2=SEARCH
...
  
```

For accessing Oracle 10g, database JDBC driver is used. This driver transparently converts the data stored in the database to and from Java strings. No special handling is necessary to access Unicode data stored in the database.

Search is implemented by `doSearch` method, which searches all property records for given criteria. If the given criteria is null or a string with 0 length, then that criteria is treated as a wildcard search. (All records will be selected for that field).

`DBStatus` object contains the result of the operation.

```

public DBStatus doSearch(String atitle, String faculty, String
department, String firstname, String lastname)
{
  
```

```

    String p_atitle;
    String p_faculty;
    String p_department;
    String p_firstname;
    String p_lastname;
  
```

```

    //Do sanity check on the parameters passed
    if((atitle == null) || (atitle.length() == 0))
  
```

```

    {
        p_atitle = "%";
    }
    else
    {
        p_atitle = atitle.replaceAll("'", "'') + "%";
    }
    ...
  
```

```

    DBStatus dbStatus = new DBStatus();
    if(conn == null)
    {
        dbStatus.flag = false;
        dbStatus.returnObject = "Database is not available";
        return dbStatus;
    }
    else
  
```

```

    {
        Statement stmt = null;
        ResultSet r = null;
        try
        {
            stmt = conn.createStatement();
  
```

```

            String query = "SELECT k.id, k.xml_data.getStringVal()
+ " FROM kadrovi k" +
            " WHERE extractValue(k.xml_data,
'/PROPERTY/LOCATION/ATITLE/text()') LIKE '" + p_atitle
+ "' AND " + " extractValue(t.xml_data,
'/PROPERTY/LOCATION/FACULTY/text()') LIKE '" +
            p_faculty + "' AND " + "
            extractValue(t.propertyxml, '/PROPERTY/LOCATION/DEPAR
TMENT/text()') LIKE '" + p_department + "' AND " + "
  
```

```

extractValue(t.propertyxml,'/PROPERTY/FIRSTNAME/text()')
>= " + p_firstname + " AND " + "
extractValue(t.propertyxml,'/PROPERTY/LASTNAME/text()')
>= " + p_lastname;
r = stmt.executeQuery(query);
if(r == null)
{
    dbStatus.returnObject = "Retrieved Null resultSet";
    dbStatus.flag = false;
}
else
{
    Vector v = new Vector();
    XMLBean xb = new XMLBean();
    xb.setPropertySchema(requestURL);
    while(r.next())
    {
        Property p = xb.getPropertyObject(r.getString(2));
        p.propertyID = r.getString(1);
        v.add(p);
    }
    if(v.size() >= 1)
    {
        dbStatus.flag = true;
    }
    dbStatus.returnObject = v;
}
}
catch(Exception e)
{
    dbStatus.flag = false;
    dbStatus.ex = e;
    dbStatus.returnObject = "An error was generated while
searching property records";
}
finally
{
    try
    {
        stmt.close();
    }
}

```

```

catch(SQLException sqe)
{
}
}
return dbStatus;
}

```

VII. CONCLUSION

As XML becomes an integrated part of today's modern businesses, the need is critical for databases, application servers, and development tools that support the family of XML standards.

Because this support is entirely standards-based, businesses are assured that their application's interoperability will be maximized.

The Oracle XML Database (XML DB) refers to the collection of XML technologies built into the Oracle Database 10g, providing high-performance and in-built storage retrieval, and processing of XML. This in-built XML functionality is integrated with the Oracle-relational database server to bridge the gap between the relational table-row and XML hierarchical storage.

Oracle's programming interfaces – XML Developer's Kit (XDK) – provide a platform to efficiently build and deploy XML solutions.

REFERENCES

- [1] N. Pitts: *XML In Record TimeTM*. Sybex Inc., 2000.
- [2] D. S. Ray, E. J. Ray, *Mastering HTML and XHTML*, Sybex Inc., 2002.
- [3] M. V. Scardina, B. Chang, J. Wang, Oracle Database 10g XML & SQL: Design, *Build & Manage XML Applications in Java, C, C++ & PL/SQL*, Berkeley, CA, McGraw-Hill/Osborne, 2004.
- [4] T. M. Robertson, *Oracle Application Server 10g, Globalization Guide 10g (9.0.4)*, Oracle Corporation, 2003.