International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:9, 2007

Use XML Format like a Model of Data Backup

Souleymane Oumtanaga, Kadjo Tanon Lambert, Koné Tiémoman, Tety Pierre, and Dowa N'sreke Florent

Abstract**—**Nowadays data backup format doesn't cease to appear raising so the anxiety on their accessibility and their perpetuity. XML is one of the most promising formats to guarantee the integrity of data. This article suggests while showing one thing man can do with XML. Indeed XML will help to create a data backup model. The main task will consist in defining an application in JAVA able to convert information of a database in XML format and restore them later.

Keywords**—**Backup, Proprietary format, parser, syntactic tree.

I. Introduction

TODAY, the numeric data became more and more crucial for enterprises and States. The recording formats of these data condition their accessibility and their interoperability. Face to the proliferation of data saving formats and their disappearance, with version and format compatibility problems, there is a worry about which format to use.

In order to guarantee an universal and non discriminatory usage of the electronic administration services and to assure the coherence of the exchanges and the interoperability of the data [1] in front of blockings generated by the proprietary systems . Henceforth, XML appears as the privileged format to exchange data between applications. Thus, it is called to play a key role of pivot format opening bridges between the diverse proposed solutions, taking into account the interoperability [2]. The purpose of the data saving model developed here, is to simplify the management of data conservation and accessibility.

Manuscript received May 9, 2007. This work was supported by the authors mentioned above.

Oumtanaga S. is with Laboratoire de Recherche en Informatique et Télécommunications (LARIT), Institut National Polytechnique Félix Houphouët Boigny, Abidjan, Côte d'Ivoire, 08 BP 475 Abidjan 08, Côte d'Ivoire (e-mail : oumtana@nic.ci, phone : 225 07900231).

Kadjo T. L. is with Laboratoire de Recherche en Informatique et Télécommunications (LARIT), Institut National Polytechnique Félix Houphouët Boigny, Abidjan, Côte d'Ivoire, 08 BP 475 Abidjan 08, Côte d'Ivoire (e-mail : nonatipv6@yahoo.fr, phone : 225 07620723)

Koné T. is with Institut de Recherches en Mathématiques (IRMA), University of Abidjan-Cocody, Côte d'Ivoire and with Laboratoire de Recherche en Informatique et Télécommunications (LARIT), Institut National Polytechnique Félix Houphouët Boigny, Abidjan, Côte d'Ivoire, 08 BP 475 Abidjan 08, Côte d'Ivoire (e-mail : tkonet@yahoo.com).

Tety P. is with Laboratoire de Recherche en Informatique et Télécommunications (LARIT), Institut National Polytechnique Félix Houphouët Boigny, Abidjan, Côte d'Ivoire, 08 BP 475 Abidjan 08, Côte d'Ivoire (e-mail : tety@nic.ci; phone : 225 05897919)

Dowa N. F. is with Laboratoire des Technologies de l'Information et de la Communication (LABTIC), Institut National Polytechnique Félix Houphouët Boigny, Abidjan, Côte d'Ivoire, 08 BP 475 Abidjan 08, Côte d'Ivoire (e-mail: flolinuxien@yahoo.fr, phone : 225 07620723).

II. Open Format and Proprietary Format

There are two groups of data saving format: Binary format also called pthereroprietary format, and Open format. A comparative study of these formats shall be made, showing their force and their weakness.

A. The Proprietary Formats

A file format is proprietary or closed if the mode of presentation of its data is opaque and its specification is not publicly available. Proprietary formats are developed by software companies in order to encode data produced by their applications: only the software produced by a company who owns the specification of a file format will be able to read correctly and completely the data contained in this file [3]. The data exchange problem appears clearly: in fact by exchanging files in proprietary formats you tacitly assume that all the recipients of your file possess the same software and the same version as you for opening the file. Any user that for technical reasons (e.g., users working on a different platform) or financial ones (users that cannot afford buying the required software) cannot run that specific software, will never be able to use the file.

There also exists a problem of data perpetuity because the survival of the user's data is bound to the editor of the format, if he disappears, the data of the user will be prisoners. Closed formats are confronted with security and confidentiality problems because during the writing of the file, the software registers numerous hidden information, often useful, but not wanted by the author [4].

B. The Open Formats

A file format is open if the mode of presentation of its data is transparent and/or its specification is publicly available. Open formats are ordinarily standards fixed by public authorities or international institutions whose aim is to establish norms for software interoperability [5]

The primary goal of open formats is to guarantee long-term access to data without current or future uncertainty with regard to legal rights or technical specification. A common secondary goal of open formats is to enable competition, instead of allowing a vendor's control over a proprietary format to inhibit use of competing products. Governments have increasingly shown an interest in open format issues [6].

Among open formats, there are format oriented presentation like RTF (introduced by Microsoft to create a standard format for text formatting), PS (The PostScript format is a language describing a page, developed by Adobe in 1985, created for printing and widely used in typography), PDF (Portable Document Format, developed by Adobe, is a format presentation document which specifications are available on the web) [5]. In all these formats, the accent is lay on the description of what you want your document to look like. Unfortunately to extract a little text from the document, you have to expurgate all the presented information [7].

There also exists the tag formats such as HTML (HyperText Markup Language) language called language of fixed tags inherited from SGML. It proposes to take care of document pagination and presentation aspect [8]. HTML is unfortunately confronted to the problem of information extraction and the problem of confusion between the presentation and the contents.

XML is also a tag format capable of resolving problems met by the other formats. Let us see together the capacities of this format to understand why it has been chosen.

C. Why is it Interesting to Save Data in XML Formats?

XML means eXtensible Markup Language. This language is extensible because contrary to HTML, XML is not a fixed format. It is a hierarchical language using text only and leaning on the Unicode format that permits the representation of text characters of all countries. XML is independent from material platforms, operational systems, and languages that use it and from protocols that transport it. This format became the pedestal of data structure notably for the Web services that permit the interoperability between incompatible systems [9]. XML simplifies the realization of the files backup which is not ambiguous and avoids the current traps, such as the non-extensibility, lack of internationalization and dependence with regard to certain platforms. It permits data to be reusable. XML format simplifies the documentary management and opens its potential of broadcasting on Internet. It facilitates the automated exchange of data between diverse systems [9].

XML takes care, not only of the presentation but also of the data content. Therefore it allows sharpen the researches in documents. That behaviour can solve problems met by search engines based on the HTML which supply impertinent results, for lack of appropriate tags to interpret the contents [9].

The structure and the organization of the data within a XML file allow the programming languages to take out data of any source to the XML format. This feature can be used to connect to a database, extract data from database and backup them in XML format, in a structured way. It is this possibility that is going to be exploited.

III. Use XML Format like a Model of Data Backup

A. Approach Developed in this Paper

The approach will consist in defining two methods:

The first one will allow extracting the data from a relational database (Mysql, SQL, Postgresql…) and save them in XML. The second one will make the opposite effect of the first one. It will restore the database by the mean of the extracted XML file.

The JAVA language will be used because of its portability. JAVA language also incorporates in several libraries for XML format managing [9].

Only the JAVA codes of these two methods are going to be exposed. Others can be easily operated by the reader.

B. In which way can this model be useful?

Database is defined as a structured collection of data items stored, controlled, and accessed through a computer.

Databases are in the heart of the organizations operations. All the vital operational information for organizations is recorded in databases. There is a plurality of database management systems with versions which don't stop to appear. The fact that these last ones save the information in an opaque way in proprietary formats does not reassure. This application was deliberately developed in JAVA to benefit from the portability of this language, and avoids the user to recompile the application. The implementation of this application is not thus complicated and does not require migrating to a particular operating system. This can be possible during the development of application interacting with the database. Indeed databases are made to be constantly used by programs. These uses can sometimes provoke the corruption or the loss of the data of the base. The backup realized by that application shall allow the restoration of the previously failing database. The restoration of a database can represent a saving solution in case of corruption or unexpected deletion. This model has the advantage to work on most of the DBMS via a JDBC driver. Thus it is enough to supply the good driver. If someone wants to do the migration towards another DBMS, the data saved in XML by the model can bring a considerable ease, because there is no need to redefine things in the new DBMS. The data of a MySQL database can be restored in a PostgreSQL base and vice-versa. Information saved in XML represents an ideal gage in term of security and assurance. XML guarantees timelessness and assures accessibility beyond any constraints of platforms of use.

All the domains of use in which this application can serve were not certainly quoted; the reader can direct it to other centres of interest.

The builders of DBMS have to think about the saving of information in standard and portable format as XML, in addition to proprietary format they propose.

IV. Development of the Applications of Backup and Restoration of the Data Base

A. The Data Backup Module

Principle: Here it is a question of defining a method in JAVA capable of connecting on a database thanks to JDBC connectors. This application is going to extract the data of this base to record them in a XML file (Fig. 1).

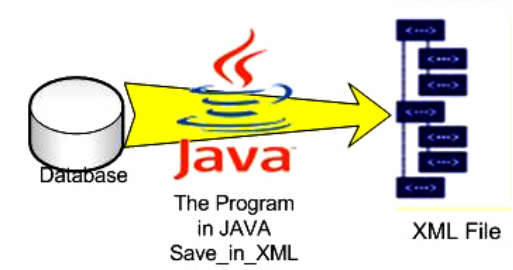

Fig. 1 Data backup process in XML format

On Fig. 1, the database is attacked by the JAVA program named Save_in_Xml(). Then the program will record the data of the database tables towards a XML file in a well structured way.

1) The document type definition

XML allows using a file to verify that a XML document is in accordance with a given syntax. The standard XML defines a document type called DTD (Document Type Definition), that is a grammar allowing verifying the correspondence of the XML document [10].

In the given DTD, the root element of the database is illustrated by the <DB> tag. A database contains 0 or several tables <TABLE>. A table has an attribute called NAME and contains 0 or several lines introduced by <ROWS> tag. In a line element (ROWS), there is one or several elements <NUMERIC> tags followed by <TEXT> tag which the NAME attributes must necessarily be specified. The DTD below is obtained in the file called ‘‘jdbc.dtd”

```
<! ELEMENT DB (TABLE)*>
        <! ELEMENT TABLE (ROWS)*>
        <! ATTLIST TABLE
          NAME CDATA #REQUIRED>
        <! ELEMENT ROWS (NUMERIC, TEXT)*>
        <! ELEMENT NUMERIC (#PCDATA)>
                <! ATTLIST NUMERIC
          NAME CDATA #REQUIRED>
          <! ELEMENT TEXT (#PCDATA)>
          <! ATTLIST TEXT
       NAME CDATA #REQUIRED>
```

2) Algorithm of data backup

The saving process needs some stages and buckles. It begins by defining the object which will be useful to receive the list of database tables, the result of the request SELECT and the meta-information about the database columns. And then it creates a PRINTWRITER object which will be necessary during the writing of XML file. If the specified table exists in the database, that table will only be added to a vector. The vector contains the list of tables. The process generates the header of the XML file called ‘‘the prologue” and also the DTD. The root tag <DB> which is important to start the database inclusion is also created. The list of the vector tables is then gone through to get back the name of each table used to write the tag <TABLE> and for the request of selection of the data from every table. The method investigates the result of the selection request such an array to write the <ROWS> tag. And then the metadata of selection request is used to get the numbers of columns. The column course allows, thanks to the methods applied to metadata, to get the type and the name of each column. According to the type of data, the method of saving fills <TEXT> or <NUMERIC> tags. So if the data type is a character, the <TEXT> tag is filled providing the name and the content. In case man has to manage data of numeric types, the <NUMERIC> tag will be filled. The whole method is contained in a block Try-Catch to print errors on the console as they occur.

3) The JAVA Code

```
public void save_in_xml(String  file, String table )
 {xmlfile= file;
  tableTosave=table;
  DatabaseMetaData dbmetadata;
  ResultSetlisttables= null;
  ResultSet selecttable= null;
  ResultSet selecttable= null;
  Vector tablecontent = new Vector();
  PrintWriter written;
  String tableName, colName,text;
  int colType,i , j=0,nbrcol,nbrtab;
  try {
    FileWriter out=new FileWriter (xmlfile);
    BufferedWriter buffout =newBufferedWriter(out);
    written=new PrintWriter(buffout);
    if (tableTosave.equals (""))
     { System.out.println ("YOU HAVE CHOSEN TO SAVE ALL THE
     TABLES OF THE DATABASE");
      dbmetadata=connection.getMetaData();
      listtables=dbmetadata.getTables(null, null, "%", null);
     while(listtables.next())
       {
     String recepttablename =listtables.getString
("TABLE_NAME");
        tablecontent.addElement(recepttablename);
        }
     listtables.close();
      } else {  tablecontent.addElement(tableTosave);  }
   nbrtab=tablecontent.size();
   if (nbrtab!=0)
    {
    written.println("<?xml version=\"1.0\" encoding=\"UTF-8\"
    standalone=\"yes\"?>");
    .written.println("<!DOCTYPE jdbc SYSTEM
    \"/home/flo/jdbc.dtd\" >");
     written.println();
     written.println("<DB>");
     for(j=0;j<nbrtab;j++)
       {
       tableName=(String)tablecontent.elementAt(j);
       written.println("<TABLE NAME=\""+tableName+"\">");
       statement=connection.createStatement();
```

```
        selecttable=statement.executeQuery("SELECT * FROM
        "+tableName);
        tablecolinfo=selecttable.getMetaData();
        nbrcol=tablecolinfo.getColumnCount();
        while (selecttable.next())
          {
          { written.println ("<ROWS>");
           for (i=1;i<=nbrcol;i++)
             {colType=tablecolinfo.getColumnType(i);
              colName=tablecolinfo.getColumnName(i);
              switch (colType)
               {
                case Types.INTEGER:
                case Types.BIGINT:
                case Types.TINYINT:
                case Types. SMALLINT:
                written.print("<NUMERIC
NAME=\""+colName+"\">");
                  written.print(selecttable.getLong(i));
                  written.println("</NUMERIC>" );
         break;
        case Types.FLOAT:
        case Types.DOUBLE:
                  written.print("<NUMERIC
NAME=\""+colNom+"\">");
                  written.print(selecttable.getDouble(i));
                    written.println("</NUMERIC>");
          break;
         case Types.VARCHAR:
         case Types.LONGVARCHAR:
                   written.print("<TEXT NAME=\""+colNom+"\">");
          text=selecttable.getString(i);
          if (text==null)
             {text="";
             }else { text=URLEncoder.encode(text); }
          written.print(text);
          written.println("</TEXT>" );
          break;
    default:    //Rien Faire
            written.println("</ROWS>" );
           }
         selecttable.close();
        written.println("</TABLE>");
       }
      written.println("</DB>");
      written.close();
      System.out.println("Sauvegarde 5/5");
     }
     }
    catch (Exception exp)
     {
     System.out.println("Erreur lors de la sauvegarde :"+exp);
     exp.printStackTrace();
     }
…..
…..
```

4) Illustration of result

number	definition
1	XML is "a flexible" way to create "common" inform...

Fig. 2 A database with Mysql

On Fig. 2, the MySql database contains a table called “my_table”. That table includes 2 properties: “number” and “definition». These two properties contain data that will be saved in XML

```
<DB>
  <TABLE NAME="my_table">
    <ROWS>
      <NUMERIC NAME="number">1</NUMERIC>
      <TEXT NAME="definition">
        XML+is+%22a+flexible%22+way+to+create+%
      </TEXT>
    </ROWS>
  </TABLE>
</DB>
```

Fig. 3 The result of data extraction in XML

On Fig. 3, there is the XML file resulting from the extraction of the data from the MySql database.

B. Module of Database Restoration

The Principle: Here starts the XML file which has been created from the database. This base is crossed by the JAVA program which extracts the data from it to restore them in the database.

Fig. 4 The database restoration process

On Fig. 4, the XML file is attacked by the JAVA program XML_Restore(). And then the program will cross the XML file and restore data towards a database in a well structured way.

1) The parser

You need a XML parser to handle a XML document. The parser loads the document into your computer memory. Once the document is loaded, its data can be manipulated using the parser you want [11].

There are several types of parser:

- SAX (Simple API for XML) is a free API that uses the events to analyse the document in XML format.
- DOM (Document Object Model) defined by the recommendations of the W3C (World Wide Web Consortium). The DOM treats the XML document as a tree. DOM is the parser that is going to be used.
- IBM supplies a free XML parser: Xerces2 Java Parser 2.8.1.

2) Algorithm of database restoration

The restoration process begins with a tree generated from the XML file. Each element of this tree is a Node. We have: base, table, column, and numeric or text data of the XML file. Each Node object can give birth to a list of affiliated node. The XML document is parsed by pulling the root node. It is here about the <DB> tags. The root <DB> allows obtaining the list of child nodes of <DB> that are the <TABLE> tags. The process recuperates then the attribute NAME of each list table. With the table name, it makes a request of deletion of the table supplied in argument or all the tables if no table had been supplied. The <TABLE> node gives birth to a list of elements <ROWS> nodes which represent the database rows table. A variable *StringBuffer* will afterward serve for beginning the construction of the request *INSERT INTO* with the tables' names. To have the columns name and their contents to form the totality of the request *INSERT*, the <*ROWS*> node is going to be exploited. The <*ROWS*> nodes engender a list of <NUMERIC> nodes and <TEXT> nodes which represent the final information contained in the database tables' rows. The restoration method gets back the attribute NAME of each element <NUMERIC> or <TEXT>. These attributes symbolize the columns of database. The program adds them to the buffer by managing commas between them. Afterward the method receives the final values between these tags. These values represent the columns contents that will be add to the buffer, managing commas. Thus SQL request INSERT is obtained in its total form. The method XML_R*estore ()* is contained in a block Try-Catch to print the likely errors. There are several methods of class Node which allow going through the syntactic tree in JAVA.

3) The JAVA code

```
public void xml_restore(String  file, String table )
  {xmlfile=file;
   tableTorestore=table;
   DocumentBuilderFactory factory;
   DocumentBuilder builder;
   Document document;
   Node db; lib_tab; lib_row; lib_num_text;
   String tableName=null,Nom_Num_Text,N_numtest;
   StringBuffer sql= new StringBuffer();
   boolean put_coma ;
   try { factory= DocumentBuilderFactory.newInstance();
       builder=factory.newDocumentBuilder();
       document=builder.parse(xmlfile);
       db=document.getDocumentElement();
       list_tab=db.getChildNodes();
       nbr_table=list_tab.getLength();




   for( i=0;i<nbr_table;i++)
     {lib_tab=list_tab.item(i);
       if(!lib_tab.getNodeName().equals("TABLE"))
          { continue;
           }else{
tableName=lib_tab.getAttributes().getNamedItem("NAME").getNo
deValue; }
}
….
```

If we want to restore all the database tables:

```
if (tableTorestore.equals (""))
  {statement.executeUpdate ("DELETE FROM "+  tableName);
   list_row=lib_tab.getChildNodes();
   nbr_row=list_row.getLength();
   for (j=0;j<nbr_row;j++)
     {lib_row=list_row.item(j);
      if(! lib_row.getNodeName ().equals("ROWS"))
        { Continue ;}
       else { sql.setLength(0);  }
      sql.append("INSERT INTO ");
      sql.append(tableName);
      for (k=0;k<nbr_num_text;k++)
         {N_numtest lib_num_text=list_num_text.item(k);
          N_numtest=lib_num_text.getNodeName();
          if((!N_numtest.equals("NUMERIC")) && (
            !N_numtest.equals("TEXT")))
           {continue;}
          if (put_coma) {  sql.append(","); }
            else{ put_coma=true; }
          if("NUMERIC".equals(lib_num_text.getNodeName()))
           {final_data=lib_num_text.getFirstChild().getNodeValue();
            sql.append(" (");
            list_num_text=lib_row.getChildNodes();
            nbr_num_text=list_num_text.getLength();
            put_coma=false;
              //lib_num_text.getNodeName();
           for(k=0;k<nbr_num_text;k++)
             if ((!N_numtest.equals("NUMERIC"))&&
              (!N_numtest.equals("TEXT")))
               { continue; }
            if (put_coma) { sql.append(","); }
              else  {put_coma=true;}
Nom_Num_Text=lib_num_text.getAttributes().getNamedItem("NA
ME").getNodeValue();
            sql.append(Nom_Num_Text);
         }
      sql.append(")VALUES (");
      sql.append( final_data);
    }
   if ("TEXT".equals (lib_num_text.getNodeName()
     {lib_num_text=list_num_text.item(k);
… …   ….
      if (final_data== null)
         { final_data="";}
       else { final_data= URLDecoder.decode(final_data);
              final_data=StringUtils.escape(final_data);
              sql.append( " \" ");
              sql.append(final_data);
             sql.append( " \" "); }
   }
 } sql.append(" )");
System.out.println(sql.toString());
rslt=rslt+statement.executeUpdate(sql.toString());}}
```

4) The illustration Result

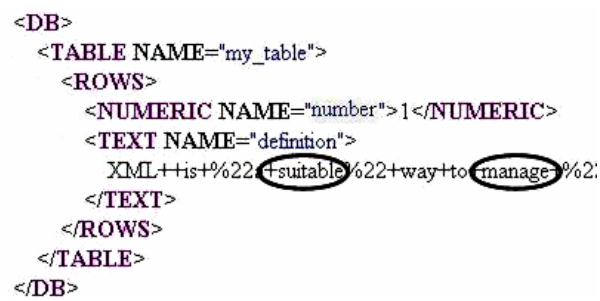

Fig. 5 A XML file

On Fig. 5, there is the XML file resulting from the previous mysql database. This file is modified by changing “flexible» into “suitable” and “create” into “manage”.

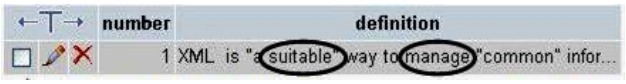

Fig. 6 The restored database

On Fig. 6, there is a MySql database resulting from the extraction of data from the XML above.

V. Conclusion

Face to the brakes resulting from blockings generated by the proprietary systems, the transparency of Open Document allows every actor or organization to exchange all the necessary documents for their functioning, with the perfect control of all the chain elements of these exchanges [12]. In no step, the contents of a document can be the object of an absence of control caused, for example, by the use of an application based on a binary format.

The use of the open format such as XML, guarantees that any document created in this format can be read or modified with the appropriate means and without limitation in the time, by the owner of this document. So, a State or a community can make a commitment towards their citizens and preserve intact the right of access to documents composing the civil, legal or administrative story of a country and his inhabitants. The perpetuity of the numeric information worries the entire world. It puts challenges at the storage level and the restoration of the electronic documents to insure the conservation and the integrity. The databases which become more and more instruments privileged for the conservation and the structuring of the data don’t escape to these anxieties.

Use XML as data backup model represents a way of drawing the attention of the DBMS builder to espouse the opened standards.

References

[1] G. Sedrati-Dinet. (2006, June 22). Des formats ouverts pour l’interoperabilité dans les administrations. [online]. Available: http://www.ffii.fr/Des-formats-ouverts-pour-l-interoperabilite-dans-les-administration

[2] J. P. Archambault.(2001, October). XML : vers un format documentaire universel ? [online]. Available: http://www.epi.asso.fr/revue/articles/a0401a.htm

[3] B. Guerry ,D. Taraborelli (2003, November 4). Why use open format? *[Online]. Available: http://www.openformats.org/main*

[4] J. P. Meyniac. (2004, May 28). Format de fichiers, attention danger ! Available : *http://www.clionautes.org*

[5] Open vs. proprietary formats, 2004. Available: http://www.openformats.org/en1

[6] Definition of open format, http://www.answers.com/topic/open-format

[7] D. Wilson ‘’Sélectionner le bon format texte’’, Gazette Linux, Janvier 2004, Vol. n°98. Available: http://ftp.traduc.org/doc-vf/gazette-linux/html/2004/098/lg98-N.html

[8] T. Boyer (2005, September 5). La norme XML. *Supinfo-Projects* Available : http://www.supinfo-projects.com/en/2005/norme_xml/

[9] M. Duperrier. (2003, February 21). XML expliqué à mon directeur général. *[Online]. Available: http://xmlfr.org/documentations/articles/030220-0001*

[10] *CommentCaMarche.* Présentation de XML. [Online] . Available : *http://abdi.pureweb.fr/telechargements/xml.doc*

[11] *XML parser. http://www.w3schools.com/xml/xml_parser.asp*

[12] OASIS ODF Adoption TC. (2006, December 10). Les avantages du format OpenDocument. [Online]. Available: http://www.oasis-open.org/committees/download.php/21451/*oasis_odf_advantages_fr_10 dec2006.odt*