

## Proposition of a model of Data backup in XML format.

SOULEYMANE OUMTANAGA<sup>1</sup>, KONE TIEMOMAN<sup>2</sup>, PROSPER KIMOU<sup>3</sup>,  
DOWA N'SREKE FLORENT<sup>4</sup>

Laboratoire de Recherche en Informatique et Télécommunication (LARIT) <sup>(1, 2, 3, 4)</sup>

Institut National polytechnique Félix Houphouët Boigny (INPHB) <sup>(1, 3, 4)</sup>

Université de Cocody, service informatique 01 BP V 34 Abidjan 01 <sup>(.2)</sup>  
08 BP 475 Abidjan 08 Côte d'Ivoire <sup>(1, 3, 4)</sup>

### Abstract

The data backup implicates the choice of formats that assure accessibility and integrity of information. Facing to these numerous data saving formats, XML is one of the simplest and most promising approaches to guarantee the perpetuity and the integrity of data. In this article, we propose a model of data backup in XML format. This model is developed with Java. Our approach consists in defining an application able to convert information of a database in XML format and restore them later. This model will allow the managing of the recurring problems of disparities between different platforms and versions met during the data saving.

**Keywords:** Portability, perpetuity, parser, URL code, Unicode, syntactic tree, node

### Introduction

All the companies produce a big quantity of crucial data which must be saved. By investing on reliable supports, they hope to protect the information from potential damages.

Besides, as well as we worry about the support, we also have to care about the format in which our data will be saved.

We remark a fit of data format today. Each data format builder uses his own format for his programs and for the same software different versions are not always compatible [1]. The question thus is to know which format can allow us to assure the portability and the perpetuity of our data.

In front of stakes of systems interoperability which crucial importance is more and more felt in front of blockings generated by the proprietary systems [9], XML stands out as the pivot of integration, opening and exchanges processes of the systems of information [8].

Our project consists in proposing a model of data saving whose purpose is to simplify the management of data conservation and accessibility.

### 1- Comparative Studies of the Data Saving Formats

Various data backup formats exist. We divide them into three (3) groups that we are going to study. We shall also find their forces and their weaknesses in order to guide us in the choice of the advantageous formats. We distinguish: The binary formats, the formats of presentation and the tag's formats

#### 1-1 The binary formats

The binary formats are format which exploitation, understanding and sometimes reading requires supplementary tools [2]. These formats are in their great majority "proprietary formats". Proprietary formats, what others call **closed formats** are developed by software companies in order to encode data produced by their applications [12]. Software like Microsoft Word in the format .DOC cannot be read by any text editor and on any platform. The problem to exchange files appears clearly [3]. If you share binary files to other user, it is not evident that this one possesses the same editor and the same version like you. It is as well very frequent as the data format builders change format for a reason or the other. But it is easy to imagine that the future version of their program will not use the same format. Your former data will be useless [2].

We remark by these aforesaid observations that the perpetuity and the integrity of the data are not guaranteed. The problem of portability was also met with the binary formats. They impose unfortunately to the users to buy a proprietary's operating system on which this format had been compiled to be able execute.

#### 1-2 the Formats of presentation

These formats allow you to see your textual information inside the document as they were originally. Examples of these types of format are RTF and PostScript [2]

The RTF format (Rich Text Format) was developed and owned by Microsoft. RTF specification is a method of encoding formatted text and graphics for easy transfer between applications

and platforms [13]. It allows to export the text by keeping (preserving) its basic shaping.

The PS (PostScript Format) was developed and owned by Adobe. It is a format of typography and pagination [13]. It was written to serve as descriptive language for texts files which would be independent from outputs peripherals.

Among the formats of presentation we can insert the PDF (Portable Document format) format. Developed by Adobe, this format has the particularity to be a binary but opened format [14]. It is presented as an improvement of PostScript. The PDF format is very used for the broadcasting of documents on Internet. A PDF file cannot be modified, but it has the advantage to conserve with exactness the presentation of document.

In all these formats, the accent is put 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 presentation information [2].

### 1 – 3 The tag’s formats

The aptness of problems of format management which influenced the data’s accessibility induces the programmers to get involved in their resolution.

It is in this context that was born the SGML (Standard Generalized Markup Language). However SGML is confronted with the problem of heaviness, complexity and data exchange [4]. Because of its presentation problems, this format is unsuitable for document writing on Internet.

HTML (HyperText Markup Language) is a language called language of fixed tags inherited from SGML. It proposes to take care of document pagination and presentation aspect [5]. HTML has evolved a lot, and it possesses a large quantity of tags. Almost all of them take interest in the presentation but they don’t give any information about the real utilisation of document content. The information extraction is very hard [2]: it blends its presentation and its structure.

XML is classified among tag’s formats and propose to solve the problems met by the aforementioned formats. We will try to give the stakes of saving data in this format.

### 2- The stakes in data saving in XML format

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 the countries of the globe. 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 System’s [8]. XML simplify the realization of the files backup which are not ambiguous, and avoid the current traps, such as the non-extensibility, lack of internationalization and the dependence with regard to certain platforms. It permits data to be reusable. XML’s format simplifies the documentary management and opens its potential of broadcasting on Internet. XML facilitates the automated exchange of data between diverse systems [8].

XML take care not only the presentation but also the data content [10]. It so allows sharpen the researches in documents. That behaviour can solve problems which search engines based on the HTML meet which supply impertinent results during the researches for lack of appropriate tags to interpret the contents [8].

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 we are going to exploit

### 3 The Saving of a Database in Xml Format

#### 3-1 Exposure of the Strategy Of Saving

Our work consists here in getting back the data of a relational database (Mysql,SQL,Postgresql...) and backup them in XML format. The data so saved can be restored at the need in the database. These two actions will be realized by two utilities which we are going to develop. Many object-oriented programming languages can offer us the possibility of arriving at this end. In our model we have chosen JAVA which is a powerful object-oriented programming languages developed by SUN. That choice is not fortuitous. The use of the set Java and XML is facilitated by the fact that they have several common points: among others independence from any platform and the management of the Unicode norms. Java incorporates a big quantity of library intended for the development with XML [9].

#### 3-2 Deployment of the Strategy

##### 3-2-1 Creation of a DTD (Document Type Definition)

A DTD is a document which contains the grammar defining the XML document. Without the DTD the document cannot be validated: We can simply verify that the syntax of the document is correct [7]. The exhaustive rules to build a DTD can be consulted on

the following Website [6]. In our case the element root of the database is illustrated by the tag <DB>. A database contains 0 or several tables <TABLE>. A Table has an attribute NAME and contains 0 or several lines <ROWS>. In an element line ROWS, we have one or several elements <NUMERIC> or <TEXT> which attributes NAME must be necessarily specified. We obtain the DTD below in the file that we call "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>
```

**3-2-2 Creation of the Module of Data Saving**

The module of saving in our model allows us to backup a relational database into XML file by means of the method *save\_in\_xml ()* of the class *eltstore* (Figure 1).

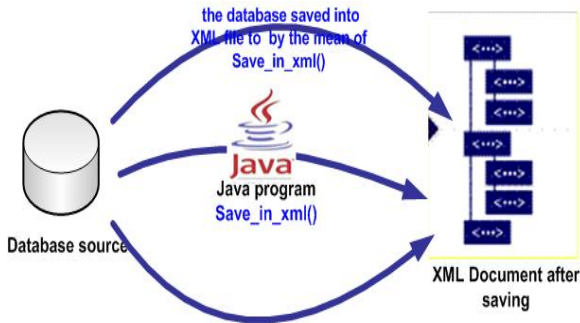


Figure 1: DATA SAVING PROCESS INTO XML FILE

On this picture we have a database, a program written in JAVA, and a document written in XML format. The database is attacked by the JAVA program *save\_in\_xml()*. And then the program will export the data of the database tables towards a XML file in a well structured way.

**3-2-2-1 Creation of the constructor of the class *eltstock ()***

The constructor receives four arguments which give information about the driver, the URL of the database, the login and password of the user for authentication. Two blocks Try-Catch are used to manage exceptions which can arise. Every object of the class must be able to be connected to the base from its creation. In case that one of these stages is ignored or badly informed, an exception is raised at once with an appropriate message.

import java.io.*;	(java.sql.SQLException
import java.sql.*;	e)

import java.net.*;	{System.out.println("Error
import java.util.*;	during the connection
import keyb.readkeyb;	or the database don't
public class eltstock {	exist !!!");
Connection connection;	try{ connection=
Statement statement;	DriverManager.getConne
String xmlfile;	ction(url,login,password);
String tableTosave	System.out.println("Good
public eltstock(String	connection to the
x try{ connection=	database "); } catch
DriverManager.getCon	(java.sql.SQLException
nection(url,login,pass	e){ System.out.println
word);	("Error during the
System.out.println("Go	connection or the
od connection to the	database don't exist !!!");
database "); } catch	System.exit(1); } }

**3-2-2-2 Definition of the method used for the saving *save\_in\_xml()***

The process of saving 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 metainformation about columns of the database. And then it creates an object PRINTWRITER which will be necessary during the writing of XML file.

If the specified table exists in the database, that table will be added only to a vector. The vector contains the list of tables. The process generates the header of the XML file called "the prologue" and the DTD. The root tag <DB> which is important to start the database inclusion is also created. The list of the tables of the vector 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 tag <ROWS>. And then the metadata of selection request is used to get the numbers of columns. The column course allows us, 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 tags <TEXT> or <NUMERIC>. So if the data type is a character, the tag <TEXT> is filled providing the name and the content. In case we have to manage data of numeric types, the tag <NUMERIC> will be filled. The whole method is contained in a block Try-Catch to print errors on the console as they occur.

public void	} else {
save_in_xml(String	tablecontent.addElement(
file, String table )	tableTosave); }
{ xmlfile= file;	nbrtab=tablecontent.size(
tableTosave=table;	);
DatabaseMetaData	if (nbrtab!=0)
dbmetadata;	{ written.println("<?xml
ResultSetlistables=	version="1.0"
null;	encoding="UTF-8"

```

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(o
ut);
écire=new
PrintWriter(buffout);
if (tableTosave.equals
("")){System.out.printl
n ("YOU HAVE
CHOSEN TO SAVE
ALL THE TABLES OF
THE DATABASE");
dbmetadata=connectio
n.getMetaData();
listtables=dbmetadata.
getTables(null, null,
"%", null);
while(listtables.next())
{String
recepttablename
=listtables.getString
("TABLE_NAME");
tablecontent.addEleme
nt(recepttablename);}
listtables.close();
written.print(selecttabl
e.getLong(i));
written.println("</NU
MERIC>"); break;
case Types.FLOAT:
case Types.DOUBLE:
written.print("<NUME
RIC NAME =\\""
+colName+"\">");
written.print(selecttabl
e.getDouble(i));
written.println("</NU
MERIC>");
break; case
Types.VARCHAR:
case
Types.LONGVARCH
AR:written.print
("<TEXT
NAME=\\""
+colName+
"\\">");
text=selectable.getStri

```

```

standalone="\yes"?>");
.written.println("<!DOCT
YPE jdbc SYSTEM
\"/home/flo/jdbc.dtd\"
>");
written.println();
written.println("<DB>");
for(j=0;j<nbrtab;j++)
{tableName=(String)table
content.elementAt(j);
written.println("<TABLE
NAME=\\""
+tableName+"\"
>");
statement=connection.cre
ateStatement();
selecttable=statement.exe
cuteQuery("SELECT *
FROM "+tableName);
tablecolinfo=selecttable.g
etMetaData();
nbrcol=tablecolinfo.getC
olumnCount();
written.println("</ROWS
>");
default:
while (selecttable.next())
{ { written.println
("<ROWS>");
for (i=1;i<=nbrcol;i++)
{colType=tablecolinfo.ge
tColumnType(i);
colName=tablecolinfo.get
ColumnName(i);
switch (colType)
{case Types.INTEGER:
case Types.BIGINT:
case Types.TINYINT:
case Types.SMALLINT:
written.print("<NUMERI
C
NAME=\\""
+colName+"\"
>");
if (text==null)
{text="";} else{
text=URLEncoder.encod
e(text);}
written.print(text);
written.println("</TEXT>
");written.println("</TAB
LE>");
written.println("</DB>");
written.close();
System.out.println("Savin
g succeed 5/5");
} selecttable.close();
} } catch (Exception exp)
{System.out.println("Erro
r during the backup
:"+exp);
exp.printStackTrace(); }

```

```
ng(i);break;
```

**III-2-3-3 Definition of the method used to close the connection close ()**

This method permits to close the SQL instruction and database connection. The method is contained in a block Try-Catch to manage the errors .

```

public void close()
{try{statement.close();
connection.close(); }
catch (Exception exp)
{System.out.println("Erro
r during the database
closure: "+exp);
exp.printStackTrace();}}

```

**3-2-3-4 Definition of the static method main ()**

In the function main we create an object of the class eltstock that will be useful in the realization of the backup. It is made in 3 phases. In the first phase we create an object providing its driver, its URL, the user name and his password. To make all these 3 phases interactive we created a class *readkeyb* that we imported previously. *readkeyb will manage the reading of* stream entered by the keyboard . In the second phase we apply the method *save\_in\_xml ()* to the object of the class *eltstock* previously created in order to accomplish the saving. We provide to the method *save\_in\_xml()*, the name of the table we want to save and the name of the file in which the table will be backup. The third step consists in calling the function *close ()* to close the object used in the end of the action of saving.

```

public static void
main(String args[]) {
keyb.readkeyb lect= new
keyb.readkeyb();
eltstock backup;
System.out.println("GIVE
THE DRIVER !!!!");
String
driver=lect.lectureString();
System.out.println ("GIVE
THE URL OF THE
DATABASE!!!!");
String
url=lect.lectureString ();
System.out.println("GIVE
THE USER LOGIN!!!!");
String
user=lect.lectureString();
System.out.println("GIVE
THE USER
PASSWORD!!!!");
passwd);
System.out.println();
System.out.println("GIVE
THE NAME OF THE FILE
IN WHICH YOU WANT TO
SAVE THE
DATABASE!!!!");
String
file=lect.lectureString();
System.out.println("GIVE
THE NAME OF THE
TABLE YOU WANT TO
SAVE OR GIVE NO NAME
TO SAVE ALL THE
TABLES OF THE BASE
!!!!");
String table=
lect.lectureString();
backup.save_in_xml
(file,table);
backup.close();}

```

**3-2-4 Creation of the Module of Database Restoration**

This Module is the opposite effect of the module of saving. The module of database restoration has the peculiarity to parse the XML file to extract the data *figure 2()*.

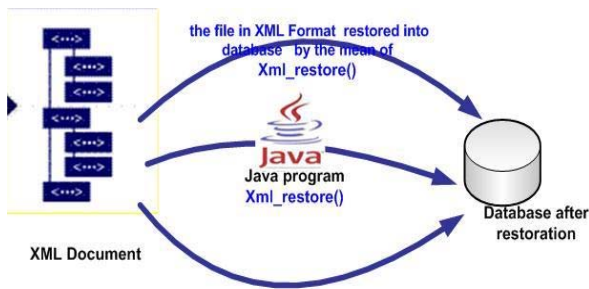


Figure 2 RESTORATION PROCESS OF XML FILE INTO DATABASE

3-2-4-1 THE CHOICE OF THE SYNTACTIC ANALYSER OR PARSER

There are several types of parser. Both more spread are those which use a tree to represent and exploit XML document and those which use events to exploit it.

SAX (Simple API for XML) is a free API that uses the events to analyse the document in XML format. The parsers which produce objects composing an arborescence to represent the XML document use the Model DOM (Document Object Model) defined by the recommendations of the W3C ( World Wide Web Consortium). For our model we opt for the model DOM. The principal task of DOM is to provide a representation of XML in memory under the shape of a tree of objects and to allow the manipulation (course, research and update) [7].

Each element that composes the tree possesses a type. With regard to this type, the element can have certain elements sons. IBM supply a free XML parser: Xerces2 Java Parser 2.8.1.This parser implements the models DOM and SAX.

III-2-4-2 Creation of the constructor of the class eltrestore ()

This constructor will receive configuration information in the form of arguments, it establish the connection to database and store the name of the XML file.

```
import java.io.*;
import java.sql.*;
import java.net.*;
import keyb. readkeyb;
import org.w3c.dom.*;
import java.util.*;
import
StringUtils.StringUtils;
import
javax.xml.parsers.*;
Connection
connection;
Statement statement;
String xmlfile;
String tableToRestore;
public class eltrestore
{public eltrestore
(String
```

```
{System.out.println("Imp
ossible to load the driver
jdbc:odbc "); }
try{connection=
DriverManager.getConne
ction(url,login,password);
System.out.println("Good
connection to the
database"); }
catch
(java.sql.SQLException
e){System.out.println
("Error during the
connection or the
database don't exist "); }
ClassNotFoundException
e)try{statement=
connection.createStateme
```

```
driverName,String
url,String login,String
password)
{ try{ Class.forName
(driverName);
} catch (
nt()); }catch
(SQLException e) {
System.out.println
("Error when preparing
the request environment
"); } }
```

III-2-4-3 Definition of the Method Used for the Database Restoration xmlrestore ()

This method parses the XML file, run through data and insert data from the file to the database. It possesses two arguments. The first one permit to receive the XML file that serves as a base for restoration, the second one represents the database table name. On the last argument we distinguish two cases. In the first case we don't specify the table name in argument. Then the program is going to save all the tables from XML files to the database. In the second case we provide the name of the table to save. And then the program searches in the file, the concerned table that will be save.

The process of restoration begins with a tree generated from the XML file .Each element of this tree is a Node. In our case as we have: the base, the table, the column, and the numeric or text data of the XML file. Each object Node can give birth to a list of affiliated node. We parse the XML document by pulling the Node root. It is here about the tags <DB>.the root <DB> allows obtaining the list of child Nodes of <DB>that are the tags <TABLE>. The process recuperates then the attribute NAME of each table of the list. With the name of tables, it makes a request of deletion of the table supplied in argument or all the tables if no table had been supplied in argument. The Node <TABLE> gives birth to a list of elements Nodes <ROWS> which represent the database table's rows. A buffer variable StringBuffer will afterward serve for beginning the construction of the request INSERT INTO with the tables names. To have the name of columns and their contents to form the totality of the request INSERT, We are going to exploit the Node <ROWS>. The Nodes <ROWS> engender a list of nodes <NUMERIC> and <TEXT> 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 columns. 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. We thus obtain our SQL request INSERT in its total form. The method xmlrestore ()

is contained in a block Try-Catch to print the likely errors. There are several methods of the class Node which allow going through the syntactic tree in JAVA.

**The beginning of the restoration program**

<pre>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_N um_Text,N_numtest; StringBuffer sql= new StringBuffer(); boolean put_coma ; try { factory= DocumentBuilderFactory .newInstance();</pre>	<pre>builder=factory.newDo cumentBuilder(); document=builder.pars e(xmlfile); db=document.getDocu mentElement(); list_tab=db.getChildNo des(); nbr_table=list_tab.getL ength(); for( i=0;i&lt;nbr_table;i++) {lib_tab=list_tab.item(i ); if(!lib_tab.getNodeNa me().equals("TABLE") ){ continue;}else{ tableName=lib_tab.get Attributes().getNamedI tem("NAME").getNode Value; }</pre>
---	--

**1<sup>st</sup> case: The Restoration of all the tables of the file XML**

<pre>if (tableToRestore.equals ("")){statement.executeUpd ate ("DELETE FROM "+ tableName); list_row=lib_tab.getChildN odes(); nbr_row=list_row.getLengt h(); for (j=0;j&lt;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&lt;nbr_num_text;k++) { N_numtest lib_num_text=list_num_text .item(k); N_numtest=lib_num_text.g etNodeName(); if(!N_numtest.equals("NU MERIC")) &amp;&amp;(!N_numtest.equals("TE XT")) {continue;} if (put_coma) { sql.append(","); } else{ put_coma=true; } if("NUMERIC".equals(lib_</pre>	<pre>=lib_num_text.getNodeN ame(); for(k=0;k&lt;nbr_num_text; k++) if (!N_numtest.equals("NU MERIC"))&amp;&amp; (!N_numtest.equals("TEX T")) { continue; } if (put_coma) { sql.append(","); } Else {put_coma=true;} Nom_Num_Text=lib_nu m_text.getAttributes().get NamedItem("NAME").get NodeValue(); sql.append(Nom_Num_Te xt); }sql.append(")VALUES ("); sql.append( final_data); } if ("TEXT".equals (lib_num_text.getNodeNa me() {lib_num_text=list_num_t ext.item(k); }){ final_data= lib_num_text.getFirstChil d().getNodeValue(); if (final_data== null) { final_data="";}</pre>
---	---

<pre>num_text.getNodeName())) {final_data=lib_num_text.g etFirstChild().getNodeValu e(); sql.append(" "); list_num_text=lib_row.getC hildNodes(); nbr_num_text=list_num_tex t.getLength(); put_coma=false;</pre>	<pre>else { final_data= URLDecoder.decode(final _data); final_data=StringUtils.esc ape(final_data); sql.append( " \" "); sql.append(final_data); sql.append( " \" "); } } } sql.append(")"); System.out.println(sql.toS tring()); rslt=rslt+statement.execut eUpdate(sql.toString());}}</pre>
--	---

**2<sup>nd</sup> case: The Restoration of the table of the XML file supplied in argument**

The program receives the name of the table supplied in argument. Except this only one aspect we have practically the same code as in the first case.

**3-2-4-4 Definition of the Method Used To Close the Connection close ()**

Here, we have exactly the same instructions already defined in the class of saving

**3-2-4-5 Definition of the static method main ()**

This method has practically the same philosophy as the one that we have in the case of the saving. To the method xml\_restore() , We supply explicitly the name of the table when we want we want to restore it alone. In case we want to restore all the tables, we don't specify the name of table. After the restoration it is necessary to release the resource with the method close ().

**4- The Interest of Database Saving Model in Xml Format**

A data base is by definition an entity in which it is possible to store data in a structured way [6].Databases have become today places of stocking of all kinds of information . We distinguish a plurality of database management systems with versions which don't stop appearing. The fact that these last ones save the information in an opaque way in proprietary formats does not reassure us. The model of data saving which we developed consists of two utilities.

The first one allows saving a relational database in the XML format. The other one restores the data from XML file to the base. This application was deliberately developed in JAVA to benefit from the portability of this language, and avoid the user to be obliged to recompile the application. The implementation of this application is not thus complicated and does not require migrating to a particular operating system. The implementation of this application on a computer can be useful during the development of application that interacts 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 our 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. These realized utilities have the advantage to work on most of the DBMS via a driver JDBC. It is thus 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 we don't 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 for us an ideal gage in term of security and assurance. XML guarantee timelessness and a sure accessibility beyond any constraints of platforms of use. We didn't certainly quote all the domains of use in which this application can serve; the reader can direct it to other centres of interest.

We remind that the objective of the exposure of this model is to propose a solution of saving information in a database, instead of a binary or proprietary format which is not sure.

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

### Conclusion

In a period where documents "paper" is more and more replaced by recordings and electronic data, the long-term access to all these data becomes a major stake. It is particularly the case of all the administrative documents which have to remain accessible and completely understandable during around ten years, even of centuries [11].

With these numerous binary formats of saving invented by numerous owners which abound and don't reassure in terms of perpetuity and integrity, XML which is a free solution positions as a sure format of data saving. Its use today does not stop growing and diversifying. The recent adoptions of XML format in the office software's (OpenOffice, Microsoft Office...) are the tangible proof [8]. The data saving model which we developed is also an edifying example of the usage of XML. Although we have not managed all the types of data, this system represents nevertheless a generic and simple tool which should for the prize of some modifications work on most of the databases accessible via a JDBC driver. The XML

file generated by a program is portable and can be used by the other programs... It is evident that such a system presents an advantage in terms of exchange between applications. XML is a technology which remains for the moment underestimated by the general public but knows some update and improvements. It is not utopian to see it becoming one day the means of transfer of Standard format between applications.

### REFERENCES

- [1] *LINUX EN ENTREPRISE : 10 Applications* STEPHANE AUBERY
- [2] *Gazette Linux n°98 SELECTIONNER LE BON FORMAT* : parut dans-Janvier 2004
- [3] <http://www.openformats.org/main> WHY USE OPEN FORMAT? : November 4 th , 2003 Bastien Guerry ,Dario Taraborelli
- [4][http://www.egshowto.com/fr/developpement/xml\\_presentation.php](http://www.egshowto.com/fr/developpement/xml_presentation.php) XML :
- [5]<http://www.supinfoprojects.com/fr/2005/norme%5Fxml/introduction/> LA NORME XML :: de Thierry BOYER Elève-Ingénieur Supinfo Océan-Indien Promotion SUPINFO 2007
- [6] [www.commentcamarche.net](http://www.commentcamarche.net) Présentation de XML :
- [7] <http://www.jmdoudoux.fr>: *Java et XML* : écrit par Jean-Michel DOUDOUX revue en 1999-2006
- [8]<http://xmlfr.org/documentations/articles/030220-0001> XML expliqué à mon directeur général : écrit par Michel Duperrier parut en 2003,
- [9] <http://www.epi.asso.fr/revue/articles/a0401a.htm> XML : vers un format documentaire universel ? : Écrit par Jean-Pierre Archambault parut en Oct 2001
- [10] <http://www.labo-sun.com> *Le XML en Java – SAX & DOM* Auteur : Stéphane Arnault
- [11] [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=office](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office) : *Les avantages du format d'OpenDocument (ODF)* Dernière mise à jour septembre 2006
- [12] <http://www.openformats.org/en1> *Open vs. Proprietary formats*: Last update 07/19/2004
- [13] <http://www.bibl.ulaval.ca/vitrine/giri> : *Guide d'Initiation à la recherche sur Internet* Edition du 1 juin 2003
- [14] <http://www.clionautes.org> *Format de fichiers, attention danger* parut le vendredi 28 mai 2004, écrit par Jean-Pierre Meyniac