XML PRESENTATION OF DOCUMENTS USED FOR DATA EXCHANGE IN MANAGEMENT OF GENETIC RESOURCES

Lina Yordanova and Vladimir Dimitrov

Abstract: In the global strategy for preservation genetic resources of farm animals the implementation of information technology is of great importance. In this regards platform independent information tools and approaches for data exchange are needed in order to obtain aggregate values for regions and countries of spreading a separate breed. The current paper presents a XML based solution for data exchange in management genetic resources of farm animals' small populations. There are specific requirements to the exchanged documents that come from the goal of data analysis. Three main types of documents are distinguished and their XML formats are discussed. DTD and XML Schema for each type are suggested. Some examples of XML documents are given also.

Keywords: XML, document's format, data exchange

Introduction

The farm animals' genetic diversity is endangered and many breeds and lines extinct every year. World Watch List [Loftuse, 1992] with endangered breeds becomes longer as well as the list of lost breeds forever. The conservation of farm animals' genetic resources needs a sustainable management of small populations in each country and region. This is connected with the establishment of information systems for data collecting, maintaining individual records for the animals and relevant data processing for population analysis.

The management of genetic resources requires separate subsystems to exchange data or to send to a centre where aggregate values to be obtained for the region or country of keeping given farm animals population. It is possible separate system nodes to use different operating systems or database management systems. This could make difficult the data exchange between them. Therefore, they need of platform independent tools what do not restrict their communications. The implementation of XML standard could be a successful approach nevertheless the target area is very complicated and there are many possible options for used documents definition.

The current work is a part of an environment for developing information system for managing small population of farm animals. The first implementation of XML standard in the environment is connected with definition of a XML format for database model and creating implementation tools for it's utilising [Yordanova, 2003].

The subject of current paper is to suggest XML formats for determined main types of documents used in the management of genetic resources. The analysis of all used documents restricts the discussion to three types:

Documents connected with data streams

Documents for data exchange in population analysis

Documents for data exchange in other kinds of data analysis.

XML Format of an Auxiliary Data Stream Document

The main type of documents exchanged in management of genetic resource is connected with data streams populating the database. A data stream is a document containing records of a same format. They could be repeated records for one animal or one record per each animal in a group. Such documents contain variety of concrete data elements and that is why their representation with a generic structure is difficult without high degree of abstraction. What we can do is to reach common XML format for the description of any data stream. If we ignore the concrete contain of the documents the result could be a very simple document tree with a root element ***stream*** and its descendent ***dataelement***. The suggested set of elements, even a simple one, will be enough for representation the structure of any document of a data stream.

The DTD of an auxiliary data stream is given in the listing 1. The root element ***stream*** is considered with an attribute for its name. The element ***dataelement*** would be well characterized with set of attributes ***name***, ***type*** and ***description***. This element could be at least once in a separate document of such type. Although it is impossible to have only one element in a document on practical reasons "once" could be accepted conditionally.

Listing 1. The DTD of XML format of a data stream

```
<DOCTYPE stream [
<!ELEMENT stream(dataelement+)>
<!ATTLIST stream                    Name        CDATA        #REQUIRED>
<!ELEMENT dataelement               EMPTY>
<!ATTLIST dataelement               Name        CDATA        #REQUIRED
                                    Type        (CHAR|HUGEINT|BIGINT|SMALLINT|
DATE|TIME|TIMESTAMP|SMALLFLOAT|BIGFLOAT|BOOL)   "CHAR"
                                    description CDATA        #REQUIRED> ]>
```

In the XML Schema of a data stream the elements ***stream*** and ***dataelement*** are defined as Complex type and the attribute ***type*** has Simple type with enumerated values.

After definition of above XML format for the description of a data stream we must discuss the way of its usage. One possibility is such XML file to be attached to the document connected with the data stream in order to describe its structure. Then the application programs of the system could use it as a dictionary for data within the stream. They also could generate a set of commands inserting the data into the database. This seems to be a generic solution applicable to all possible data streams with different structure.

As a second possibility we consider the conversion of the data stream documents to the XML format that must be a solution of a separate information system. A separate XML format reflecting the structure of a given document could be developed and implemented there.

An example of a XML file containing a description of a data stream is given in listing 2. This one describes the data stream named Semen from the information system "Cryo" [Groeneveld, 2002].

Listing 2. A XML file with description of a data stream

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE stream SYSTEM "file:///home/lina/teza/potok.dtd">
<stream name="Semen">
   <dataelement description="The external id for the bull"
             name="ext_animal"/>
   <dataelement description="The external id for the reporting unit"
             name="unit"/>
   <dataelement description="The date of semen delivery"
             name="delivery_dt" type="DATE"/>
   <dataelement description="The date of semen production"
             name="production_dt" type="DATE"/>
   <dataelement description="The number of doses total"
             name="no_doses" type="SMALLINT"/>
   <dataelement description="The number of straws per dose"
             name="no_straws" type="SMALLINT"/>
   <dataelement description="The type of straws"
             name="type_straws"/>
   <dataelement description="Quality mark- motility post freezing %"
             name="mot_post" type="SMALLINT" />
   <dataelement description="Quality mark- after collection %"
             name="mot_after" type="SMALLINT" />
   <dataelement description="Semen certificate identification"
             name="certificate_id" type="SMALLINT"/>
</stream>
```

The XML file from the listing 2 is created and validated through defined DTD with XML editor what could be done also by any program application working with XML standard.

The documents of a data stream could be obtained on various approaches. Nevertheless which approach is used the documents of data streams have the logical structure defined above. The defined XML format of their description can be used in document exchange. In consequence data from corresponding XML document of a data stream could be inserted into the database via middleware software.

There are practical cases in management of genetic resources with manual filling in paper documents and than converting in any electronic form. The most common situation is conversion to a comma separated value format. Technical tools produce files of the same format often. It is possible also that the concrete document to be converted to its own XML format.

The documents of data streams are connected in general with the GUI forms for inserting and manipulating data [Yordanova, 2000]. The GUI forms in the environment are created according to them. The description of a data stream does not contain the access actions to the database elements in order to insert data from the stream.

We should consider that the data streams are mainly connected with primary data collecting. It is very seldom the data they contain to be retrieved from another database but if this is the case then the approaches from next chapters are applicable.

XML Formats of Documents for Population Analysis

The data exchange in management of genetic resources covers different groups of data depending on the purpose of their analysis.

The population analysis needs of data about the animal origin. Such analysis requires obtaining of individual inbreeding coefficients, effective population size and other genetic parameters that the manager of breed conservation program could choose. Then the data exchange between the center and peripheral nodes must include: the identification of the animal, the identifications of its parents, gender and birth date or birth year. This set of data is a minimum, enough for calculation the genetic parameters for population analysis.

We define the XML format of document containing data for animal origin and its individual identification via DTD (listing 3) and XML Schema.

Listing 3. The DTD of a document for data exchange in population analysis

```
<DOCTYPE pedigree [
<!ELEMENT pedigree(animal+)>
<!ATTLIST pedigree                  name    CDATA    #REQUIRED>
<!ELEMENT animal(birthdt, sire, dam)>
<!ATTLIST animal                    ext_id  CDATA    #REQUIRED
                                    unit    CDATA    #REQUIRED
                                    gender  (F | M)  #REQUIRED>
<!ELEMENT birthdt(#PCDATA)>
<!ELEMENT sire(#PCDATA)>
<!ATTLIST sire                      ext_id  CDATA    #REQUIRED
                                    unit    CDATA    #REQUIRED>
<!ELEMENT dam(#PCDATA)>
<!ATTLIST dam                       ext_id  CDATA    #REQUIRED
                                    unit    CDATA    #REQUIRED> ]>
```

The root element is called ***pedigree*** and its sub element is ***animal***. The sub element is defined to be at least once in the document. It has attributes ***ext_id***, ***unit*** and ***gender*** and sub elements ***birthdt***, ***sire*** and ***dam***. The elements ***sire*** and ***dam*** should have the same attributes like the element ***animal***. In all cases the attribute ***ext_id*** means an external identification of an animal depending on the unit that reports the animal. Here it is not convenient to use for ***ext_id*** type ID, because in common case it is not unique. Two units can report two animals with the same identification. That requires the couple of elements (***ext_id, unit***) to be unique.

The element or attribute connected with information about the breed to which the animal belongs is not included. This is done because the population analysis supposes that the data collecting concerns animals from the same breed. If it is necessary one could mark the breed into the name of the document or to add an element breed. For the given example the breed name is stored in the attribute ***name*** of the root element (listing 4). The example document is for population analysis according the XML definition explained above. The data is for two family couples. Four animals (the progeny of the families) are included. The document is checked for validation with corresponding XML schemas through program applications that use DTD or XML Schema.

Listing 4. An example for data exchange in population analysis

```
<?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE pedigre SYSTEM "file:///home/lina/teza/pedigre.dtd">
 <pedigre name="minipigs">
   <animal ext_id="1677" gender="F" unit="12">
         <birthdt>23.12.1998</birthdt>
         <sire ext_id="3456" unit="12">04.04.1997</sire>
         <dam ext_id="2345" unit="12">12.03.1996</dam> </animal>
   <animal ext_id="1698" gender="F" unit="12">
         <birthdt>23.12.1998</birthdt>
         <sire ext_id="3456" unit="12">04.04.1997</sire>
         <dam ext_id="2345" unit="12">12.03.1996</dam> </animal>
   <animal ext_id="1701" gender="M" unit="12">
         <birthdt>3.11.1998</birthdt>
         <sire ext_id="5003" unit="12">12.08.1996</sire>
         <dam ext_id="4312" unit="12">12.03.1996</dam> </animal>
   <animal ext_id="1702" gender="F" unit="12">
         <birthdt>3.11.1998</birthdt>
         <sire ext_id="5003" unit="12">12.08.1996</sire>
         <dam ext_id="4312" unit="12">12.03.1996</dam> </animal>
   </pedigre>
```

XML Formats of Documents for Other Kinds of Analysis

The common structure for documents exchanged in the management of genetic resources with the goal to perform other kinds of analysis contains mandatory animal identification and its one or multytrait measurements. A possible generic XML scheme of such document is given in the listing 5 with DTD.

Listing 5. The DTD of a document for data exchange in other kinds of analysis

```
<DOCTYPE data [
<!ELEMENT data(animal+)>
<!ELEMENT animal(trait+)>
<!ATTLIST animal              ext_id  CDATA      #REQUIRED
                              Unit    CDATA      #REQUIRED>
<!ELEMENT trait(measurement+)>
<!ELEMENT measurement EMPTY>
<!ATTLIST measurement         Date    CDATA      #REQUIRED
                              Value   CDATA      #REQUIRED
                              Type    CDATA      #REQUIRED> ]>
```

The root element named *data* is consisted by at least one sub element *animal*. It has attributes *ext_id* and *unit*s as well as one sub element trait meet more than once. The element *trait* connects any investigated trait to an animal. It is possible a document to have data about more traits that complete a process. One animal could be measured many times for a trait. That is why it is appropriate to have a sub element *measurement* repeated many times. The measurement is characterized with attributes or sub elements *date*, *value* and *type*. The measurements type requires from the application programs to maintain with external coding for different types of measurements.

The animal identification is given with attributes *ext_id* and *unit*, which means maintaining external identification for both objects, *animal* and *unit*. This requires from the software to obtain a new or to retrieve existing internal identification from the database. The last one is used according the system supporting unique identification for the animals. About reporting unit it is most possible to have only second situation.

A XML document obtained on the defined format is given in the listing 6. The document is validated according the XML format definition. It contains weight measurements of two animals.

Listing 6. An example of XML document with data for the trait weight

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE data SYSTEM "file:///home/lina/teza/data.dtd">
<data>
   <animal ext_id="6768" unit="11">
       <trait name="weight">
            <measurment date="12.03.2000" type="bdw" value="0.3"/>
            <measurment date="12.05.2000" type="wdw" value="0.6"/>
            <measurment date="14.06.2000" type="rw" value="1.9"/>
            <measurment date="13.07.2000" type="us" value="2.4"/>
       </trait>
   </animal>
   <animal ext_id="4546" unit="11">
       <trait name="weight">
            <measurment date="14.04.2000" type="bdw" value="0.2"/>
            <measurment date="15.06.2000" type="wdw" value="1.7"/>
            <measurment date="14.07.2000" type="us" value="2.6"/>
       </trait>
   </animal>
</data>
```

For the documents of data streams and for the data exchange it is preferable to get data automatically. Retrieving data from the database is common for the documents for data exchange. Very seldom in small populations' management data is generated automatically from computer systems with measure tools. Then generated files are usually in CSV format. Their transformation to the XML format is not a difficult task. Better solution is the XML format defined for a data stream to be used as a meta form of such CSV file. The middleware software can operate with data according the description of the document structure in XML format.

Middleware Software for Data Exchange

In current work the XML documents are used for data transfer after generating or parsing by dynamic program applications. The main processes discussed here are:
Parsing and retrieving data from a XML document in order to be inserted into the database
Creating a XML document retrieving data from the database.

The related program codes use the XML formats of the documents in data exchange defined above.

The program code for inserting data into the database from a XML document

Let the XML document that is going to be parsed and analyzed in order to insert data into the database has the structure from listing 3. Let the target database has the conceptual database scheme for small populations management [Yordanova, 2000], where the main relations are named TRANSFER, UNIT and ANIMAL. Then the algorithm for parsing the documents and inserting the data into the database includes the next steps:

```
Begin                         (#begin
DBI connection                (# Connection to the database
Objects and variables         (# Declaration of objects and variable
SQL statements                (# Definition of SQL statements -
Foreach $row                  (# For each animal retrieving of:
  Sire/ID, sire/unit            (# db_sire from TRANSFER(ext_id, unit)
  Dam/ID, dam/unit              (# db_dam from TRANSFER(ext_id, unit)
  Animal/unit                   (# db_unit from UNIT via unit
  get_next_val(sequence_name)   (# new db_animal identification
  INSERT into ANIMAL, TRANSFER  (# Inserts in TRANSFER and ANIMAL
end foreach                   (# End of the cycle
Db disconnect                 (# Disconnect the database
End                           (#End
```

If any parent does not have internal database identification then a new one has to be obtained in the relation TRANSFER and recorded into the relation ANIMAL. The inserts will not be done if there is another animal with the same values of (ext_id, unit). The released program is a Perl code and uses the module XML::XPath of Matt Sergeant that implements the XPath standard and allows fast search and parsing elements of XML document via a tree of document's nodes.

The program code for retrieving data from database

The function of the code is connected with the XML format for population analysis (listing 3). The algorithm is separated to two main steps:

1. Getting via Query a set of tuples, containing the external identification of all active animals and their parents as well as their reporting units, birth dates and gender.
2. Recording data from the tuples in XML document elements.

This code uses Perl&XML module XML::Writer that allows creating of XML document via defined objects.

Conclusion

The usage of XML standard makes the data exchange in management genetic resources much more flexible and platform independent. There are a lot of program applications that work with many operating systems and could facilitate implementation of defined XML formats of documents for data exchange. The user could create the XML documents containing the description of data streams using: 1) XML editors that apply XML declarations DTD or XML Schema; 2) program applications that includes processing and validating XML documents through schema.

The other XML documents for data exchange in all kinds of data analysis for small populations could be created and used via briefly presented here program codes.

The defined XML formats could be extended and could become a base language for data exchange in management of genetic resources.

Bibliography

[Groeneveld, 2002] E. Groeneveld, L.Yordanova and S.J. Hiemstra. An Adaptable Management Support System for National Genebank, 7th World Congress on Genetics Applied to Livestock Production, August 19-23, 2002, Montpelier, France, Session 26, Management of genetic diversity, 513-516, 2002.

[Loftuse, 1992] R. Loftuse and B. Scherf, World Watch List for Domestic Animal Diversity, FAO Rome, 1992.

[Yordanova, 2000] L. Yordanova, E. Groeneveld. The implementation Strategy of Information System with Existing Data Streams, Vortrastagung der Deutschen Gesellschaft fuer Zuchtungkunde e. V. (DGfZ) u. der Gesellschaft fur Tierzucht Wissenschaft (GfT), A28, 2000.

[Yordanova, 2003] L. Yordanova, E. Groeneveld, Vl. Dimitrov. A XML Approach for Support DB Modeling. In: Proceedings of the Thirty Second Spring Conference of the Union of Bulgarian Mathematicians, 297-302, 2003.

Authors' Information

Lina Yordanova – Thracian University, Department of Informatics, Mathematics and Physics, Stara Zagora, 6000, Studentsko gradtche, Bulgaria; e-mail: lina@uni-sz.bg

Vladimir Dimitrov – Sofia University, Faculty of Mathematics and Informatics, Sofia, Bulgaria; e-mail: cht@fmi.uni-sofia.bg