IFLA
2005
OSLO

World Library and Information Congress:
71th IFLA General Conference and Council

"Libraries - A voyage of discovery"

August 14th - 18th 2005, Oslo, Norway

Conference Programme: http://www.ifla.org/IV/ifla71/Programme.htm

Июнь 8, 2005

Code Number: 064-E
Meeting: 121 UNIMARC with Information Technology

UNIMARC XML Slim Schema: Living in new environment

Vladimir Skvortsov

Standing Member of IFLA
Permanent UNIMARC Committee,
Head of the National Service on
RUSMARC Format Development,
National Library of Russia,
Saint-Petersburg, Russia
vskv@nlr.ru

Olga Zhlobinskaya

Senior Researcher,
National Library of Russia,
Saint-Petersburg, Russia
olga_zhlobinskaya@nlr.ru

Alla Pashkova

Senior Programmer,
National Library of Russia,
Saint-Petersburg, Russia
alla@nlr.ru

Abstract

The paper discusses the role of XML and its perspectives in library information systems, particularly with regards to basic functions of bibliographic formats – storage and transportation of the data. Slim XML Schema for UNIMARC representation is presented, its main features being lossless conversion from MARC to XML, roundtripability from XML back to MARC, support for embedded fields and extended range of indicator values, independence from any specific dialect of MARC format, stability to any changes of the format.

Obviously we live in the time when no specialist might isolate himself within his field of activity, without knowing about what is going on beyond that field. Perhaps in everyday life indeed – our neighbours' affairs are their own business only, but in scientific and business world this is not the case, and here a curious cat does not have to worry about his life. Definitely – XML is not an immediate business of library community, but in fact we can see now this interest at the highest level of library community – at the World Library and Information Congress.

So this paper deals with UNIMARC XML-Slim schema. The schema, which we are going to talk about, was developed by the Centre for Computer Technologies of the Ural State University (Ekaterinburg) in collaboration with the National Service on RUSMARC format development.

Strictly speaking, "slim schema" is called "slim" just because the schema itself does not require long and detailed examination, which, however, does not belittle its merits anyway. It's exactly like in people's world – the most stout person is not necessarily the most important one, and vice versa. The same is true about XML-schemas. However, to understand why XML-Slim schema is important, one should have a clear view at the place of the Schema in the whole XML-environment. In other words, before we could speak on merits of the Schema, we should understand why it is necessary in general and what we might do with it.

Let us try to start ab ovo, in the hope that the most advanced listeners would excuse us. The point is that, as the practice shows, it happens sometimes that quite good, high-level report on important achievements in a some new field of knowledge – results in pure emotional impression: anyone can feel intuitively that it is really good, but it would be better to understand what is the point and where we can use it.

We would take a risk to suppose that the audience of the World Library and Information Congress includes not only experts in the field of XML.

So,

First of all, we should understand – what is XML?

XML (eXtensible Markup Language) is a simplified dialect of the SGML language, designed to describe hierarchical data structures in World Wide Web. It has been developed by W3C since 1996; the most current specification is XML 1.0 (Third Edition, 2004).

XML is a simple markup language, describing arbitrary structured data. Strictly speaking, it is not a language in the true sense, but rather a meta-language for defining specialised languages describing specifically structured data. Unlike HTML, XML may contain any tags, which are considered to be necessary by the authors of XML-vocabulary.

In particular, it is evident, that any existing MARC format might be presented in XML environment.

But, is it really necessary?

Here it might be useful to remind what is the point of existing MARC formats? Strictly speaking, two main functions of a MARC format are storage and transportation of the data.

Storage

Despite apparent simplicity, XML has quite complicated mechanisms to control correctness of the data (i.e. to validate the data), it allows to verify hierarchical links in the document, and, what is the most important, determines common standard for the documents where the data is stored, whatever the nature of the data could be.

Can we consider XML and related techniques as a real database, i.e. as a data base management system? The answer is: “it’s something like that”. On the one hand, XML allows to implement many things that we can find in “normal” database: storage (XML-documents), schemes (DTD, XML-scheme language), query languages (XQuery, XPath, XQL, XML-QL, QUILT, etc.), program interfaces (SAX, DOM, JDOM), etc. On the other hand, among disadvantages of XML one can mention lack of many possibilities, which present in real data bases: efficient storage, indices, safety, transactions and data integrity, multi-user access, triggers, cross-document queries, etc.

On the other hand XML-documents can be stored in so-called “native XML database”. This term was originally used for Tamino database from Software AG. Native XML database is much like other database; the main difference is that internal structure of native XML database is based on XML rather than any other model (e.g. relational model).

One of main reasons to store data in native XML database is rate of access. Depending on how a native XML database stores data in terms of physics, it can access data much faster than any of relational databases. The reason is that some techniques of storage, used in native XML databases, store documents physically as a whole, or use physical (rather than logical) pointers between parts of the document. This allows to extract documents either without necessity to combine them at all, or using physical combination. Any way of the two is much faster than logical combination, which is used in relational databases.

Besides, XML-documents can be stored in relational database as well. The choice depends on the structure of a document, and there are a lot of works nowadays dealing with a strategy of the choice.

Transportation

Nowadays main container to transport data in MARC format is ISO 2709. As compared to ISO 2709 XML has a number of obvious merits.

1. XML does not have some limitations of ISO 2709. Syntax of ISO 2709 was designed according to the needs of the technological environment of its time and does not completely correspond to requirements of modern technology. For example, in ISO 2709 the length of record is restricted to 99999. As a result, information, which is evidently bibliographic one (detailed abstract or comprehensive notes), in some cases can not be put into the MARC-record. In XML length of the record is not limited. Also ISO 2709 does not provide place for binary (non-textual) data, such as image of the book cover or any accompanying material - graphic, audio or video. XML allows doing such kind of things.
2. XML-document is humanreadable.

   At least in the sense that one can analyse XML document de visu without the problems he would have trying to do it on ISO 2709 record.
3. One of important advantages of XML-documents is the fact that in spite of relatively easy way of creating and processing (one can use any text processor to edit XML documents and standard XML parser to process them), XML document provides technique to create structured information which might be easily “understood” by computer.
4. XML has quite strict syntax, which makes it possible to formalise control during the document creation.
5. XML supports hierarchical relations, so it is possible to reflect hierarchical structure of bibliographical records.
6. XML-document may contain information on layout of the data when displayed or printed; such information might be presented using the language for style description

(XSL).

7. Important merit of XML is its native integration with web-interfaces, and consequently – support by manufacturers of certain software, such as Web-browsers.

All the above brings us to consider XML as de facto a standard for data exchange via Web-interface.

It seems to be enough for now to justify the enthusiasm of bibliographic community regarding XML we mentioned above.

OK, we want XML, but what does it mean?

What do we need to have to say we've got XML?

The set we suggest below appears not to be something mandatory, this is just an example of what could provide necessary storage and transportation of bibliographic data in XML environment.

We must have at least two XML-documents:

1. XML Slim Schema

   Slim (or Transport) Schema defines MARC formats at the most common level and in this sense it plays the same role as ISO 2709 does. Slim Schema thus is to serve as a base for normative schemas we are going to tell about below.

   It should be presented on Web, and any interested party should know the address.

   Any Normative Schema contains a reference to Slim Schema to indicate what is its basis.

   Slim Schema should be quite simple and enable lossless conversion to ISO 2709 and back to XML. Major characteristic of the Slim Schema is its steadiness for any changes in MARC formats. Slim Schema enables correct structuring of the data, but not semantic correctness.

2. XML Normative Schema.

   Normative Schema is in fact the MARC format converted to XML.

Slim and Normative Schemas are the two major elements of our set.

3. Besides, this set could be supplemented by:

- utilities for conversion ISO 2709 – XML and XML – ISO 2709;

- tools to validate XML-document for conformity to slim schema;
- tools to validate XML-document for conformity to normative schema;
- stylesheets for displaying records in the user-friendly form (tag – content);
- tools to validate bibliographic record using XSLT-transformations.

Now when we recognize what we need let us go back to our Slim schema

Today the most widely used and the most well-known slim schema is MARC21XML/Slim by the Library of Congress. However there are some problems in using this Schema for UNIMARC unlike ISO 2709, which does not have those problems.

In particular, the Schema does not make provisions for marking up embedded fields, which are widely used in UNIMARC and in some national formats of UNIMARC-family; next, the range of indicators values in UNIMARC is wider than MARC21XML/Slim admits. That is why it was important to work out mechanism for representing embedded fields etc. in XML – both for lossless conversion and for adequate representation of syntax and semantics of such record.

So, when developing XML Slim Schema for UNIMARC our task was: using MARC21XML/Slim and keeping all its basic elements and principles, to develop XML Schema which could be used for any of existing MARC-formats. The Schema (Fig. 1) was developed as it was already mentioned by the Centre for Computer Technologies of the Ural State University (Ekaterinburg) in collaboration with the National Service on RUSMARC format development.

Main design principles for the Schema:

- lossless conversion of MARC-record in ISO 2709 to XML-document;
- validation of record (checking correctness of the record structure) with XML-parser;
- taking into account specific characteristics of UNIMARC format;
- highest possible independence from any specific dialects of MARC format;
- stability to changes of UNIMARC format;

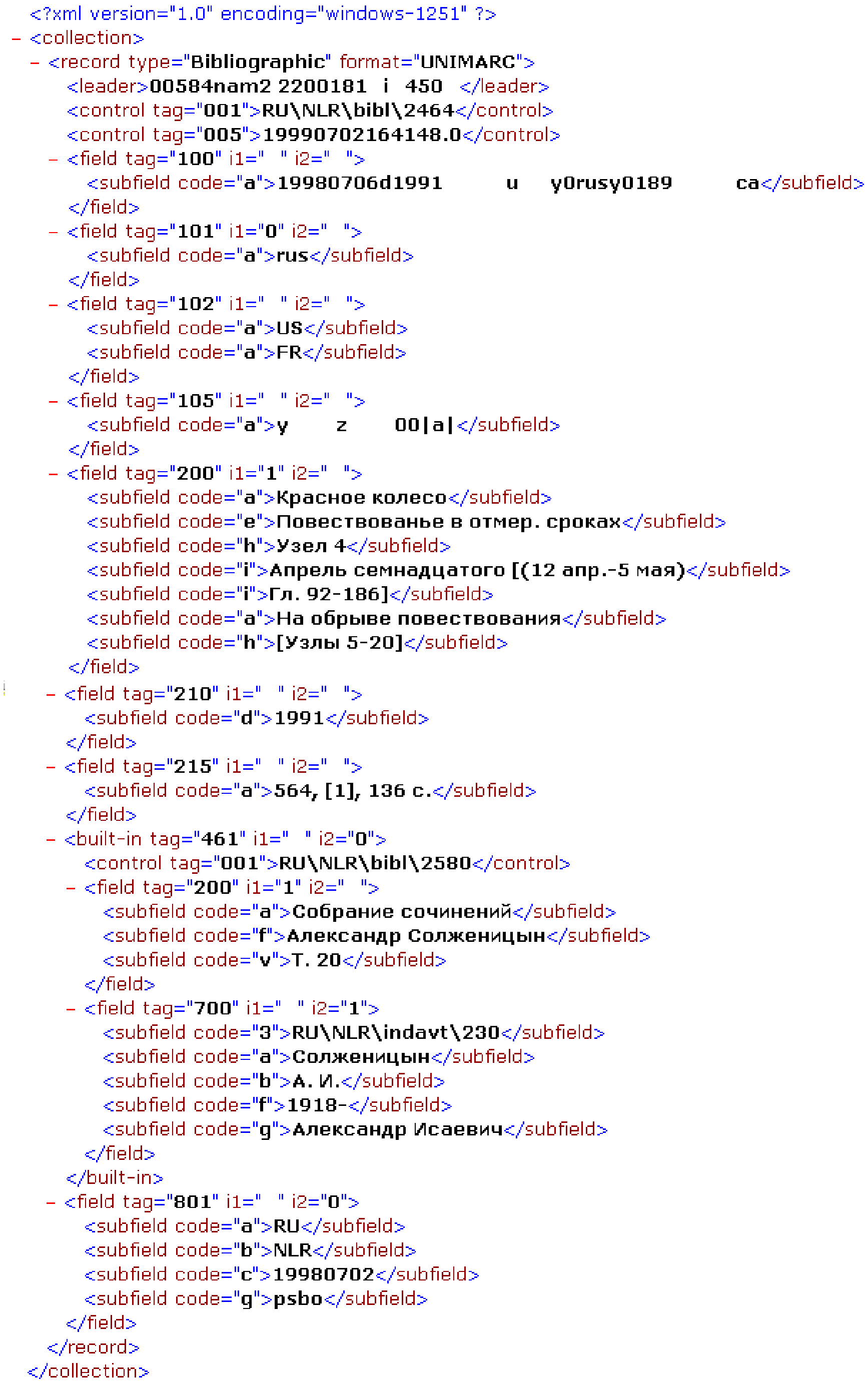
You can see the result of our work below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="collection">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="record" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="record" type="recordType"/>
  <xs:complexType name="recordType">
    <xs:sequence>
      <xs:element name="leader" type="leaderType"/>
      <xs:element name="control" type="controlType"
      maxOccurs="unbounded"/>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="field" type="fieldType"/>
        <xs:element name="built-in" type="built-inType"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="type" use="optional">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="Bibliographic"/>
          <xs:enumeration value="Authority"/>
          <xs:enumeration value="Holdings"/>
          <xs:enumeration value="Classification"/>
          <xs:enumeration value="Community"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="format" use="optional">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="UNIMARC"/>
          <xs:enumeration value="RUSMARC"/>
          <xs:enumeration value="MARC21"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
  <xs:simpleType name="notEmptyString">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="leaderType">
    <xs:restriction base="xs:string">
      <xs:pattern value="\d{5}[ 0-9A-Za-z]{5}22\d{5}[ 0-9A-Za-z]{3}450[ 0-
      9A-Za-z]"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="controlTag">
    <xs:restriction base="xs:string">
      <xs:pattern value="00[1-9a-zA-Z]"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="controlType">
    <xs:simpleContent>
      <xs:extension base="notEmptyString">
         <xs:attribute name="tag" type="controlTag" use="required"/>
         <xs:attribute name="id" type="xs:ID" use="optional"/>
         <xs:attribute name="idref" type="xs:IDREF" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="fieldTag">
    <xs:restriction base="xs:string">
      <xs:pattern value="(0[1-9][0-9])|([1-9][0-9]{2})"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="fieldType">
    <xs:sequence>
      <xs:element name="subfield" type="subfieldType"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="tag" type="fieldTag" use="required"/>
    <xs:attribute name="i1" type="indicatorType" use="required"/>
    <xs:attribute name="i2" type="indicatorType" use="required"/>
    <xs:attribute name="id" type="xs:ID" use="optional"/>
    <xs:attribute name="idref" type="xs:IDREF" use="optional"/>
  </xs:complexType>
  <xs:simpleType name="indicatorType">
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="preserve"/>
      <xs:pattern value="[\da-z \|]"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="built-inType">
    <xs:choice maxOccurs="unbounded">
      <xs:element name="control" type="controlType"/>
      <xs:element name="field" type="fieldType"/>
    </xs:choice>
    <xs:attribute name="tag" type="fieldTag" use="required"/>
    <xs:attribute name="i1" type="indicatorType" use="required"/>
    <xs:attribute name="i2" type="indicatorType" use="required"/>
    <xs:attribute name="id" type="xs:ID" use="optional"/>
    <xs:attribute name="idref" type="xs:IDREF" use="optional"/>
  </xs:complexType>
  <xs:simpleType name="subfieldIdentifier">
    <xs:restriction base="xs:string">
      <xs:pattern value="[\da-zA-Z]"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="subfieldType" mixed="true">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:any processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="code" type="subfieldIdentifier" use="required"/>
    <xs:attribute name="id" type="xs:ID" use="optional"/>
    <xs:attribute name="idref" type="xs:IDREF" use="optional"/>
  </xs:complexType>
</xs:schema>
```

Fig. 1. UNIMARC XML Slim Schema

Figure 2 shows example of bibliographic record in UNIMARC using UNIMARC XML Slim Schema.



```
  <?xml version="1.0" encoding="windows-1251" ?>
- <collection>
  - <record type="Bibliographic" format="UNIMARC">
      <leader>00584nam2 2200181  i  450  </leader>
      <control tag="001">RU\NLR\bibl\2464</control>
      <control tag="005">19990702164148.0</control>
    - <field tag="100" i1="  " i2="  ">
        <subfield code="a">19980706d1991       u    y0rusy0189       ca</subfield>
      </field>
    - <field tag="101" i1="0" i2="  ">
        <subfield code="a">rus</subfield>
      </field>
    - <field tag="102" i1="  " i2="  ">
        <subfield code="a">US</subfield>
        <subfield code="a">FR</subfield>
      </field>
    - <field tag="105" i1="  " i2="  ">
        <subfield code="a">y      z      00|a|</subfield>
      </field>
    - <field tag="200" i1="1" i2="  ">
        <subfield code="a">Красное колесо</subfield>
        <subfield code="e">Повествованье в отмер. сроках</subfield>
        <subfield code="h">Узел 4</subfield>
        <subfield code="i">Апрель семнадцатого [(12 апр.-5 мая)</subfield>
        <subfield code="i">Гл. 92-186]</subfield>
        <subfield code="a">На обрыве повествования</subfield>
        <subfield code="h">[Узлы 5-20]</subfield>
      </field>
    - <field tag="210" i1="  " i2="  ">
        <subfield code="d">1991</subfield>
      </field>
    - <field tag="215" i1="  " i2="  ">
        <subfield code="a">564, [1], 136 с.</subfield>
      </field>
    - <built-in tag="461" i1="  " i2="0">
        <control tag="001">RU\NLR\bibl\2580</control>
      - <field tag="200" i1="1" i2="  ">
          <subfield code="a">Собрание сочинений</subfield>
          <subfield code="f">Александр Солженицын</subfield>
          <subfield code="v">Т. 20</subfield>
        </field>
      - <field tag="700" i1="  " i2="1">
          <subfield code="3">RU\NLR\indavt\230</subfield>
          <subfield code="a">Солженицын</subfield>
          <subfield code="b">А. И.</subfield>
          <subfield code="f">1918-</subfield>
          <subfield code="g">Александр Исаевич</subfield>
        </field>
      </built-in>
    - <field tag="801" i1="  " i2="0">
        <subfield code="a">RU</subfield>
        <subfield code="b">NLR</subfield>
        <subfield code="c">19980702</subfield>
        <subfield code="g">psbo</subfield>
      </field>
    </record>
  </collection>
```

Fig. 2. UNIMARC record in XML

This XML Slim Schema, unlike MARC21XML/Slim, indeed covers any MARC formats, because being based on MARC21XML/Slim it also takes to account the embedded fields and extends the range of indicators values according to provisions of UNIMARC.

You can find it at http://www.rba.ru/rusmarc/soft/rusmarc_slim.xsd.