UNIMARC'S Embedded Fields and MarcXchange. Unexpected Plots

Vladimir Skvortsov,
National Library of Russia.
St. Petersburg,
Russia.
vskv@nlr.ru

I would like to talk on XML Slim in the light of ISO/DIS 25577, which is currently under development. The standard is intended to be an analogue of the ISO 2709 container in XML environment. The way which the developers chose to solve the problem, i.e. direct emulation of the ISO 2709 to XML – seems to be absolutely evident and not promising any surprises.

But, as you can see from the title of the presentation, it's going to deal with unexpected plot, or, to be more precise, with unexpected plots; it means that we are going to touch at least two plots, consequences of which are difficult to predict right now.

1. **First plot – embedded fields**

As you know, UNIMARC, equally with technique of standard subfields provides for using special constructions, which are called embedded fields.

Below you can see an example of such a construction:

451#0$1001BLN6956090**$1**2001#$aPrefaces to the experiences of literature
**$1**210##$aNew York$cHarcourt Brace Jovanovich$d1979

Embedded fields are placed to the ISO 2709 container exactly as standard fields.

Using the above example: it is supposed that we have standard field 451 with the only repeatable subfield $1; all the data included in the $1 subfield thus being an object of semantic analysis inside automated library systems and it does not have any relation to syntax of ISO 2709.

It seems natural to suppose that when syntax of ISO 2709 is translated to that of XML – the situation should be absolutely the same without any unexpected consequences.

Let us see for example how embedded fields would look like in XML SLIM schema.

With kind permission of Tony Curwen I will address to the example given in one of his messages, and reproduce two questions he had asked in connection with this example.

```
<datafield tag="451" ind1=" " ind2="0">
  <subfield code="1">001BLN6956090</subfield>
  <subfield code="1">2001 $aPrefaces to the experiences of
  literature</subfield>
  <subfield code="1">210  $aNew York$cHarcourt Brace
  Jovanovich$d1979</subfield>
</datafield>
```

What is there to identify 001, 200 and 210 as tags, and the following digits or spaces as indicator values?

(I note also that section 4 refers to IS2, Field separator, and IS3, Record separator, but makes no mention of IS1, Subfield delimiter).

I find his questions essential and deliberately chose not to invent my own examples to emphasize that certain doubts concerning the approach being used by ISO Technical Committee 46 – by no means have subjective nature. I mean direct transferring of ISO 2709 syntax to that of XML.

I would like to note as well that the above example illustrates the both plots, which are to be subject of my presentation. But first let us do with the first of them.

Answer for the first question

– What is there to identify 001, 200 and 210 as tags, and the following digits or spaces as indicator values? –

Like with ISO 2709 – we have quite sufficient data for semantic analysis: subfield $1 shows that next 7 symbol positions (except the field 001) should be treated consequently as the tag of the field (3 symbols), indicators (2 symbols) and subfield delimiter (2 symbols), which is followed by the data itself.

But, if we try to answer the above question in terms of XML syntax

– What is there to identify 001, 200 and 210 as tags, and the following digits or spaces as indicator values? –

The answer is to be quite simple – nothing.

So what is the problem? Nothing changes. It is still now like it was before.

But why do we change ISO 2709 to XML? Unlike ISO 2709 where data loading and indexing were executed by internal tools of the automated system, XML provides standard and, what is important, external means for these operations. Thus, if you do not use embedded fields, your

system, working in XML, might (ideally) have no special tools for data processing during its loading and indexing.

But if you do work with embedded fields, XML would give you just partial solution, and you would have to think on purchasing or developing special software for semantic analysis, which in this case is necessary for data loading and indexing. Besides, your system, working in XML, would have to treat properly the links specified in the embedded fields.

It should be noted, that in this case the tools that would provide proper work – are certainly non-standard ones, since that is the basic question under discussion – whether we should introduce such kind of tools into the standard or not. Today the standard does not include such tools.

This means, in particular, that no one standard automated system could support these tools. And, as we told above, if your library or library community, or even the whole country works with embedded fields – it is your own problem.

I am afraid that under such circumstances the very technique of the embedded fields has little chance to survive – and this is quite probable consequence of the decision which is close to be approved today as ISO 25577.

Well, do we really need embedded the fields as such? With your kind permission, I will try to pose the attitude to embedded fields which I believe to exist today: those who do not use them, probably, consider them of no special necessity; but those who realized the opportunities given by using embedded fields in practice – would hardly want to abandon using them.

This is apart from huge financial and technological problems, which would arise in case of rejection of the embedded fields for those who worked with them.

To do with the first plot I would like to summarize the above considerations.

Direct and immediate transferring of ISO 2709 syntax to that of XML would certainly give us guarantee in the sense of providing possibility of converting transport schemes to each other. But it would also mean that we do not need any progress (excuse me for a sort of industrial terminology) in the field of container production.

From our point of view, the question on the embedded fields is in fact the question on the linking technique – and this is the thing which ISO 2709 was not definitely supposed to do.

Consequently, XML Slim must be some new step in developing of transport schemas not only in the sense of language development. On the other hand this step should not be done in the way to loose their round-trippability. Below we will show that this is quite possible.

Now let's look to the second plot.

2. **Second plot – subfield delimiters**

As you might remember, that was the second question from Tony Curwen. If we look at his example again we will see that in embedded fields subfield delimiters are contained in the body of XML-document. As we know, symbol of dollar '$' is used just for graphic record representation. As for a transport schema, hexadecimal code 1F (or IS1) is used as subfield delimiter.

Now let us see at W3C Recommendation regarding to characters, which might be used in a text string of any XML-document.

Character Range

[2] `Char` ::= `#x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFD] | [#x10000-#x10FFFF]`

You can see here that 1F code goes straight before the range of acceptable codes: `[#x20-#xD7FF`, and so 1F is not included in the range of characters recommended by W3C to be used in the body of XML-documents.

It means, in particular, that nowadays using embedded fields in XML is completely unacceptable.

All the above was supposed to provoke interest and anxiety first of all on the part of those who work with embedded fields. I also hope for interest and anxiety of Permanent UNIMARC Committee and UNIMARC Core Activity on the whole.

The rest of our respected audience could have sufficient grounds to feel detached onlookers of progress the above plots. To change the situation and involve this part of the listeners in our second plot, I will mention a field, which I believe to be sometimes forgotten even by its own creators.

In terms of structure this field differs from fields of the 4-- block of UNIMARC being discussed above, but in fact it is embedded field by its very nature. This field is present both

in UNIMARC, and in MARC21, where, in fact, it has come to UNIMARC from. The field we are talking about is 886 DATA NOT COVERED FROM SOURCE FORMAT (UNIMARC) and 886 Foreign MARC Information Field (MARC21).

That is how the field looks like in UNMARC and MARC21 formats:

> **8862#$2**ukmarc**$a**083**$b**00$aRussia. Education$b- Biographies - Collections
> (Example from UNIMARC Manual)
>
> **8862b≠2**ukmarc**≠a**690**≠b**00≠a00030≠dGreat Britain≠z11030≠abatterflies
> ≠z21030≠alife cycles
> (Example from USMARC Manual)

So, in XML Slim Schema it should look like this:

```
</datafield>
  <datafield tag="886" ind1="2" ind2=" ">
    <subfield code="2">ukmarc</subfield>
    <subfield code="a">083</subfield>
    <subfield code="b">00$aRussia. Education$b- Biographies - Collections</subfield>
  </datafield>

</datafield>
  <datafield tag="886" ind1="2" ind2="b">
    <subfield code="2">ukmarc</subfield>
    <subfield code="a">690</subfield>
    <subfield code="b">00≠a00030≠dGreat Britain≠z11030≠abatterflies≠z21030≠alife cycles
      </subfield>
  </datafield>
```

As we can see, in this case subfield delimiters are also contained in the body of the XML-document. It means that today the 886 field falls out of any transport XML schema both for UNIMARC and for MARC21 format.

Now, after the sketch in such dark tones has been painted, it is the time to speak about a possible solution.

3. Solution

At the UNIMARC session at World Library and Information Congress in Oslo in August 2005 we proposed XML Slim Schema, covering any MARC formats, working with embedded fields and non-conflicting with subfield delimiters described above in the second plot.

Besides, like ISO 25577, the Schema would provide for possibility of round-trippability of transport schemas.

Since our presentation at Oslo we brought the Schema to conformity with the draft of ISO 25577, keeping none the less all the possibilities provided by the Schema. And now we can state proudly that our Schema keeps all the merits of the draft ISO 25577 while having no its shortcomings.

This does not mean that we created anything extraordinary. The solution, as you can see, is quite simple and rather evident. Moreover, it is not the only possible one: I mean, for example, the Schema, which was proposed by Giovanni Bergamin.

Our XML Schema can be seen on the Web:
http://www.rba.ru/rusmarc/soft/UNISlim.xsd

Using our Schema the example from Tony Curwen's letter would look like as follows:

```
<built-in tag="451" ind1=" " ind2="0">
        <control tag="001">BLN695609</control>
        <datafield tag="200" ind1="1" ind2=" ">
                <subfield code="a">Prefaces to the experiences of
                 literature</subfield>
        </datafield>
        <datafield tag="210" ind1=" " ind2=" ">
                <subfield code="a">New York</subfield>
                <subfield code="c">Harcourt Brace Jovanovich</subfield>
                <subfield code="d">1979</subfield>
        </datafield>
</built-in>
```

The Schema defines additional nesting level, corresponding to any embedded field. Following UNIMARC, we define the only possible nesting level, but in principle nothing prevents us from defining any number of such levels, as it is done, for example, in the Schema, proposed by Giovanni Bergamin.

This example shows part of the record, where all tags could be recognized and unambiguously interpreted by XML-parsers. And such solution, as it was told above already, - is not the only possible one. Now it is turn for organizational decisions.

Nonetheless, they are such kind of decisions, which often turn out to be the most difficult things.