A case for XTM 3.0

Alexander Mikhailian[1], Rani Pinchuk[1], and Xuân Baldauf[2]

[1]Space Applications Services NV – Belgium
{ami,rp}@spaceapplications.com

[2]University of Auckland, New Zealand
xuan--xtm3--2008--tmra.de@academia.baldauf.org

Abstract. Improvements to XTM 2.0 are suggested in this paper. First, a set of criteria is defined for evaluating those improvements. It is followed by the suggestions themselves: align element names with the names used in TMDM, reduce the number of elements by introducing mixed content and using attributes whenever it is possible. Finally, some relevant irregularities are discussed.

XTM 2.0[1], The recent standard for Topic Maps exchange is an important step in the direction of popularization of Topic Maps. However, it evolved from the legacy XTM 1.0 [2] format and is lacking clarity at many points. We list below a number of possible changes to XTM 2.0 that have become apparent during the use of XTM 2.0 in our day-to-day work.

The proposed changes to XTM 2.0 will help achieve the following goals:

1. **Make the format more compact.** XML by itself is quite verbose, so care should be taken not to worsen the situation.
2. **Improve the parsing speed.** The speed criterion does not need further explanation.
3. **Simplify the parser development.** By simplification we mean reducing the number of parsing rules.
4. **Improve the readability.** Although XTM is rather machine-readable, occasional reading of XTM documents by humans for debugging and learning purposes should be possible.
5. **Improve the learning curve.** Developers should be able to understand XTM easily and quickly, with minimized risk of misconceptions.

We will start incrementally, going from the simple, self-evident improvements to the more critical ones.

Maicher, L.; Garshol, L. M. (eds.): *Subject-centric computing. Fourth International Conference on Topic Maps Research and Applications, TMRA 2008, Leipzig, Germany, October 16-17, 2008, Revised Selected Papers.* (Leipziger Beiträge zur Informatik: XII) - ISBN 978-3-941152-05-2

Each proposed change will be summarized with regard to the declared goals.

1 Align element names

1.1 rename itemIdentity

Element names in XTM 2.0 map unambiguously to those in TMDM. However, there is one element that has a slightly different name in XTM 2.0 than the related term in TMDM. It is `itemIdentity`, which is called *item identifier* in TMDM. It costs nothing to bring it back in line with TMDM.

from XTM 2.0:

```
<itemIdentity href="#shakespeare-wrote-hamlet"/>
```

to XTM 3.0:

```
<itemIdentifier href="#shakespeare-wrote-hamlet"/>
```

There is another case of naming inconsistency. XTM 2.0 uses just `name` for what TMDM calls *topic name*, but this can be justified, as this element is a child of the `topic` element. The dependence of the `name` on the `topic` is thus expressed by extralinguistic means.

This change allows to improve on goals 4 and 5.

2 Reduce the number of elements

2.1 introduce mixed content in topic names

In XTM 2.0, the element `name` contains the element `value` which in turn contains the text as `#PCDATA`. The element `value` has no meaning in itself, as it just allows to avoid mixed content. While this made sense at the time when XML processing tools were not mature enough, there is less reason not to use mixed content nowadays, when issues surrounding the mixed content have been widely discussed and understood [4]. We may then remove the `value` element.

Note that the whitespace handling rules for mixed content are not different from those for text content. While editing mixed content by hand, a human editor may be tempted to insert carriage returns and spaces without taking into account the fact that those carriage returns and spaces will be carried on as is by the XML

parser. Fortunately, XTM is not supposed to be directly modified by humans, except for debugging and illustration purposes, as in this paper.

from XTM 2.0

```
<name>
  <value>Shakespeare's authorship of Hamlet</value>
</name>
```

to XTM 3.0

```
<name>Shakespeare's authorship of Hamlet</name>
```

This change allows to improve on goal 2. In general, this change also improves on goal 4, except for the cases where the mixed content is actually mixed, that is, where the content contains `type`, `scope` *or* `variant` *elements. Later in the paper, we will convert the* `type` *and the* `scope` *elements into attributes, leaving only the* `variant` *element. Thus, this change improves on goal 4 except when there are* `variant` *elements. Because variants are a rarely used feature in topic maps, we believe that this change is a general improvement on goal 4.*

2.2 remove the `topicRef` element

The `topicRef` element has two slightly different usages. In one usage, it appears as a mandatory child of the `type` element or the `role` element and it may be thought of as a superfluous envelope for the `href` attribute. In the other usage, groups of `topicRef` elements appear as children of `scope` and `instanceOf` elements, each `topicRef` element providing an envelope for the `href` attribute.

In both cases, the parent elements `type`, `role`, `scope` and `instanceOf` indicate the affected property and the `href` attribute determines the value of the property. We may thus drop the `topicRef` element without affecting the data model:

from XTM 2.0

```
<type>
  <topicRef href="#written-by"/>
</type>
...
<scope>
  <topicRef href="#history-of-literature"/>
  <topicRef href="#authorship-issue"/>
</scope>
...
<role>
```

```
  <type>
    <topicRef href="#author"/>
  </type>
  <topicRef href="#shakespeare"/>
</role>
```

to XTM 3.0

```
<type href="#written-by"/>
...
<scope href="#history-of-literature"/>
<scope href="#authorship-issue"/>
...
<role href="#shakespeare">
   <type href="#author"/>
</role>
```

The element `type` under `role` is mandatory, which allows us to convert it into an attribute. We may also rename the reference to the *association player* from `href` into a more mnemonic `player` attribute:

to XTM 3.0

```
<type href="#written-by"/>
...
<scope href="#history-of-literature"/>
<scope href="#authorship-issue"/>
...
<role player="#shakespeare" type="#author"/>
```

This change is positive for all goals.

2.3 introduce mixed content in variants

The `variant` element can either contain a reference or inline data. This is translated into XTM 2.0 through two elements, `resourceRef` and `resourceData` that can alternatively appear below `variant`. A slightly more compact notation would alter the possible contents of the `variant` element depending on whether we want to use a reference or to paste inline data. The definition of the `variant` element in Relax-NG would then be as follows:

```
variant = element variant {
  (href, reifiable, scope+) | (reifiable, scope+, text)}
data = element data { datatype?, any-markup}
```

And the actual XML would change as follows:

from XTM 2.0

```
<variant>
  <scope>
    <topicRef href="#wikipedia"/>
  </scope>
  <resourceData>Shakespeare authorship question
                </resourceData>
</variant>
<variant>
  <scope>
    <topicRef href="#wikipedia"/>
  </scope>
  <resourceRef
href="http://en.wikipedia.org/wiki/Shakespeare_authorship"/>
</variant>
```

to XTM 3.0

```
<variant>
  <scope href="#wikipedia"/>Shakespeare
  authorship question</variant>
<variant
  href="http://en.wikipedia.org/wiki/Shakespeare_authorship">
  <scope href="#wikipedia"/>
</variant>
```

This change allows to improve on goal 2, as well as on goal 4, see section 2.2 for the details.

2.4 introduce mixed content in occurrences

The same reduction of the `resourceRef` and `resourceData` elements can be applied for the `occurrence` element. We will as well convert the `type` element into an attribute.

```
<occurrence type="#wikipedia"
 href="http://en.wikipedia.org/wiki/Shakespeare_authorship"/>
```

Just as the previous change, this one allows to improve on goals 2 and 4.

3 Simplify the association

3.1 use attributes whenever possible

We have already started to bring the complex hierarchy of elements under the `association` element to a very compact form by using attributes whenever possible. Let us make the final step and convert the *association type* into an attribute, as well.

from XTM 2.0

```
<association reifier="#shakespeare-wrote-hamlet">
  <type>
    <topicRef href="#written-by"/>
  </type>
  <role>
    <type>
      <topicRef href="#author"/>
    </type>
    <topicRef href="#shakespeare"/>
  <role>
  </role>
    <type>
      <topicRef href="#work"/>
    </type>
    <topicRef href="#hamlet"/>
  </role>
</association>
```

to XTM 3.0

```
<association reifier="#shakespeare-wrote-hamlet"
             type="#written-by">
  <role player="#shakespeare" type="#author"/>
  <role player="#hamlet" type="#work"/>
</association>
```

This change has a major positive effect on all goals.

4 Relevant irregularities

4.1 the `instanceOf` controversy

Until now, we have tried to make the expression of a topic map in an XML document shorter, hoping that a concise representation will bring along readability and will allow for an easier parsing. However, there are cases where the simplification makes for a verbose output.

There is a notorious exception to the way associations are encoded in XTM 2.0. A `type-instance` association can be encoded as a shortcut in the form of an `instanceOf` element. This special case may be unfolded into an `association` element, which would allow to drop the `instanceOf` element from the format.

from XTM 2.0

```
<topic id="shakespeare-wrote-hamlet">
  <instanceOf>
    <topicRef href="#academic-debate"/>
  </instanceOf>
```

to XTM 3.0

```
<association type="#type-instance">
  <role player="#academic-debate" type="#type"/>
  <role player="#shakespeare-wrote-hamlet" type="#instance"/>
</association>
<!--declarations of topics are skipped-->
```

The arguments around the `instanceOf` element are numerous. The summary table below lists several of those:

in favour of `instanceOf`	*against* `instanceOf`
It is by far the most used association type and deserves a special treatment.	It requires implicit knowledge and hardens the learning curve.
Allows for shorter XML and for faster parsing.	Increases the complexity of the parser.
Provides better readability.	Inconsistent with `supertype-subtype` association type.

The most popular argument in favor of the `instanceOf` element is related to the frequency of its use. The well known Italian Opera [5] topic map contains 1826 `type-instance` associations and only 31 `supertype-subtype` associations. This is a decisive argument. We will *retain* the `instanceOf` element.

from XTM 2.0

```
<topic id="id1">
  <instanceOf>
    <topicRef href="#academic-debate"/>
  </instanceOf>
  <name><value>...</value></name>
</topic>
```

to XTM 3.0

```
<topic id="id1">
  <instanceOf href="#academic-debate"/>
  <name>...</name>
</topic>
```

Abandonment of `instanceOf` *has not been proposed.*

4.2 Controversy around `itemIdentity`

The section 3.6 of TMDM [3] states that the `item identifier` is a

> ...locator assigned to an information item in order to allow it to be referred to.

It has a twofold purpose, and serves as the identifier for the topic map constructs, as well as a way to trace back the origins of the topic map construct, created by merge. Tis further explained in the section 5.1 of TMDM [3]:

> In a sense item identifiers are identifiers for topic map constructs, but unlike subject locators and identifiers devoid of any specified semantics. Item identifiers may be freely assigned to topic map constructs.
> One specific use of item identifiers is in the deserialization from the XML syntax where item identifiers are created that point back to the syntactical constructs that gave rise to the information items in the data model instance.

It is not defined whether the locator is local to the topic map only or universal. However, the reference to the URI [6] and IRI [7] standards for locators in TMDM implies that the universal addressing is at least possible, if not required.

On the other hand, the section 6.2 of [3] explains that during the merging of two topics A and B, a new topic C is created with its `item identifiers` property set

> ...to the union of the values of A and B's item identifiers properties.

This leads to a contradiction that is better explained by the following example of merge of the topics. Let us consider two topic maps, *A* and *B*:

Topic map *A* with the IRI `uri://base1/`

```
<topicMap version="2.0">
 <topic id="id1">
   <subjectIdentifier href="http://www.tmra.de/2008/"/>
</topic>
</topicMap>
```

Topic map *B* with the IRI `uri://base2/`

```
<topicMap version="2.0">
 <topic id="id1">
   <subjectIdentifier href="http://www.tmra.de/2008/"/>
</topic>
</topicMap>
```

Both of these topic maps are merged into a new topic map *C* with the IRI `uri://base3/`.

Topic map *C* with the IRI `uri://base3/`

```
<topicMap version="2.0">
 <topic id="id1">
   <subjectIdentifier href="http://www.tmra.de/2008/"/>
   <itemIdentifier href="uri://base1/#id1"/>
   <itemIdentifier href="uri://base2/#id1"/>
 </topic>
</topicMap>
```

Before merging, there existed exactly one topic with the item identifier `uri://base1/#id1` (the topic in topic map *A*). After merging, however, there exist two topic items with the item identifier `uri://base1/#id1` (the topic in topic map *A* and the topic in topic map *C*). Thus, the item identifier is not universal, anymore. Or, in other words, it can not be addressed from outside of a topic map.

Such a constraint contradicts TMDM [3] in that it effectively enforces a scope on the item identifier which TMDM does not have. It also leaves without any foundation the use of the IRI [7] standard for encoding item identifiers in XTM 2.0 [1].

We solve the contradiction by enforcing the *one topic – one item identifier* principle. We propose that topics have at most one item identifier. When merging two topics *a* and *b* into a new topic *c*, the new topic *c* should get a new item identifier distinct from the item identifiers of *a* and *b*.

Next to the addressing, the second use of item identifiers in XTM 2.0 [1] is to track the origins of a topics. In order to keep this functionality, we introduce a new *item origins* property. This property shall be set to the union of item identifiers of the topics that contributed to the merging. A new topic shall have its *items origins* set empty. A topic created by merging should have its *item origins* property set to the union of item identifiers of the contributing topics.

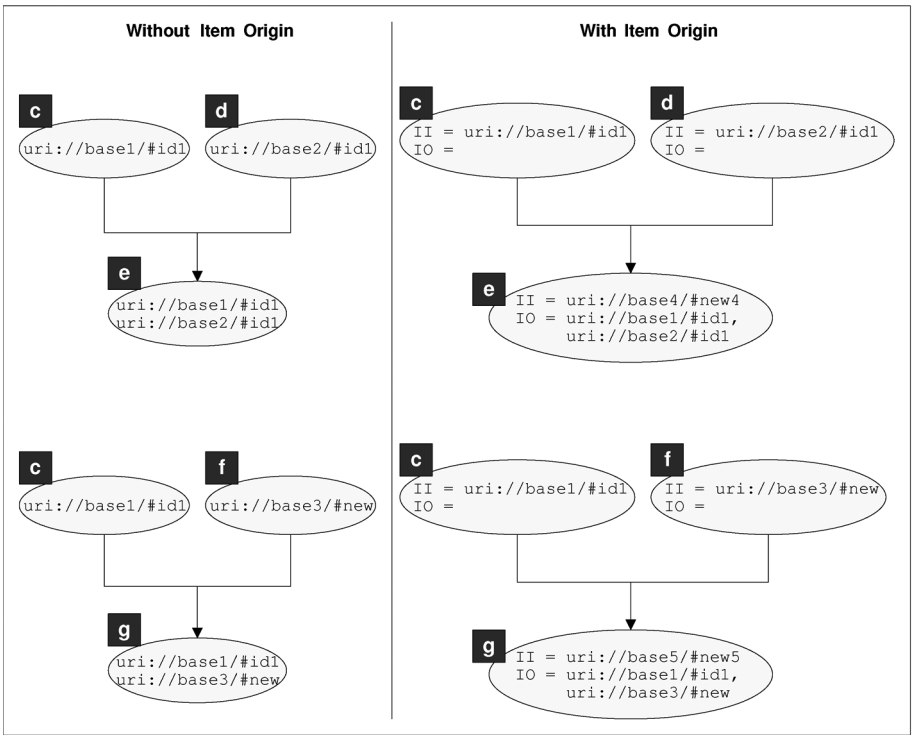

Fig. 1: A more complex use case

The advantage of having *item origins* can be further exemplified by the use case presented in Fig. 1. In this figure, each oval represents a topic from a different topic map. The text inside the topics on the left part represents the *item identifiers* of the topics. On the right side, `II` stands for *item identifier*, and `IO` stands for *item origin*. The arrows between the topics represent merging. For example, the topic *c* is merged with the topic *d* and the result is topic *e*.

As we can see on the left side of the figure, merging *c* and *f* results in the topic *g*. This topic has two item identifiers. One of them is `uri://base1/#id1`. If we try to find the origin of the topic *g* according to this item identifier, we will find that

it can be either the topic *c* or the topic *e*. However, the topic *g* did not originate from the topic *e*.

On the right side, we can clearly identify the origin of the topic *g*, due to the introduction of the item origins.

Implications for XTM 3.0. Because the item identifier of a topic item can be unambiguously determined by the `id` attribute of its `topic` element, we can drop the element `itemIdentifier` altogether. Instead, we introduce the `itemOrigin` element to contain the *item origin* property. The topic map *C* will thus look as follows:

Topic map *C*

```
<topicMap version="3.0">
 <topic id="id1">
   <subjectIdentifier href="http://www.tmra.de/2008/"/>
   <itemOrigin href="uri://base1/#id1"/>
   <itemOrigin href="uri://base2/#id1"/>
 </topic>
</topicMap>
```

This change positively impacts the goals 3, 4 and 5.

4.3 Ensure completeness

Now that the concepts of *item origin* and *item identifier* have become separate, we are able to use the `xsd:ID` data type for encoding item identifiers and `xsd:IDREF` to point to them. This way, the completeness of the document can automatically be verified at the XML parser level.

Note that using identifiers of type `xsd:ID` has a further advantage. Each `scope` element currently serves just as container for referencing a topic. We replace the list of `scope` elements per statement by a `scope` attribute of that statement. This is possible, because a list of `xsd:IDREF` values, one for each `scope` element, can be represented by one XML attribute of type `xsd:IDREFS`. Such an attribute will contain a list of `IDREF` values separated by spaces. Consider the following example:

from XTM 2.0

```
<topic id="tmra2008">
  <name>
    <scope>
      <topicRef href="#english"/>
      <topicRef href="#y2k-pbl"/>
```

```
    </scope>
    <type>
      <topicRef href="#short-name"/>
    </type>
    <value>TMRA'08</value>
  </name>
  <name>
    <value>TMRA 2008</value>
  </name>
</topic>
```

to XTM 3.0

```
<topic id="tmra2008">
  <name type="short-name"
        scope="english y2k-pbl">TMRA'08</name>
  <name>TMRA 2008</name>
</topic>
```

This change positively impacts the goals 3, 4 and 5.

5 Conclusion

Not all the goals set at the beginning of the paper can be objectively evaluated. For instance, the readability of the XTM 3.0 documents and a flatter learning curve may only be confirmed by users once the format starts to gain acceptance. The easiness of the parser development shall be evaluated on the actual parser code, coming preferably from multiple implementations.

There is however a way to measure the compactness and, indirectly, the parsing speed by comparing the size of the XTM 3.0 file to the size of the XTM 2.0 file containing the same data. A test on the Italian Opera [5] topic map shows a twofold decrease in the size of the XTM 3.0 document with regard to the XTM 2.0 document.

6 Acknowledgement

This work has been partly funded by the Flemish government through the IWT/ITEA2 project LINDO (ITEA2-06011).

References

1 ISO/IEC IS 13250-3:2007: Information Technology - Document Description and Processing Languages - Topic Maps XML Syntax. International Organization for Standardization, Geneva, Switzerland. http://www.isotopicmaps.org/sam/sam-xtm/

2 XML Topic Maps (XTM) 1.0 v 1.16 2001/08/06 14:31:44 http://www.topicmaps.org/xtm/

3 ISO/IEC IS 13250-2:2006: Information Technology – Document Description and Processing Languages – Topic Maps - Data Model. International Organization for Standardization, Geneva, Switzerland. http://www.isotopicmaps.org/sam/sam-model/

4 Sean McGrath: Mixed content myopia. http://www.itworld.com/nl/xml_prac/07112002/

5 Italian Opera Topic Map. http://www.ontopia.net/operamap/

6 RFC 3986, Uniform Resource Identifiers (URI): Generic Syntax, Internet Standards Track Specification, January 2005 http://www.ietf.org/rfc/rfc3986.txt

7 RFC 3987, Internationalized Resource Identifiers (IRIs), Internet Standards Track Specification, January 2005, http://www.ietf.org/rfc/rfc3987.txt

A A sample XTM 3.0 file

```
<topicMap xmlns="http://www.topicmaps.org/xtm/" version="3.0">
  <topic id="shakespeare-wrote-hamlet">
    <subjectIdentifier href="#shakespeare-wrote-hamlet"/>
    <instanceOf ref="academic-debate"/>
    <name scope="wikipedia">Shakespeare's
      authorship of Hamlet<variant>Shakespeare
      authorship question</variant>
    </name>
    <occurrence
      href="http://en.wikipedia.org/wiki/Shakespeare_authorship"
      type="wikipedia"/>
  </topic>
  <association reifier="shakespeare-wrote-hamlet"
    type="written-by"
    id="shakespeare-wrote-hamlet-association">
    <role player="shakespeare" type="author"/>
    <role player="hamlet" type="work"/>
  </association>
  <topic id="wikipedia">
    <name>Wikipedia</name>
  </topic>
  <topic id="written-by">
    <name>Written by</name>
  </topic>
  <topic id="shakespeare">
    <name>William Shakespeare</name>
  </topic>
  <topic id="author">
    <name>Author</name>
  </topic>
  <topic id="hamlet">
    <name>Hamlet</name>
  </topic>
  <topic id="work">
    <name>Work</name>
  </topic>
  <topic id="academic-debate">
    <itemOrigin href="iri://abstract-topics/#academic-debate"/>
    <name>Academic deabate</name>
  </topic>
</topicMap>
```

B The RelaxNG schema

```
default namespace = "http://www.topicmaps.org/xtm/"
namespace xtm = "http://www.topicmaps.org/xtm/"
datatypes xsd = "http://www.w3.org/2001/XMLSchema-datatypes"

start = topicMap

href = attribute href { xsd:anyURI }
ref = attribute ref { xsd:IDREF }
id = attribute id { xsd:ID }
reifiable = attribute reifier { xsd:IDREF }?, itemOrigin*
datatype = attribute datatype { xsd:anyURI }
version = attribute version { "3.0" }
type = attribute type { xsd:IDREF }
player = attribute player { xsd:IDREF }
scope = attribute scope { xsd:IDREFS }

itemOrigin = element itemOrigin { href }
subjectLocator = element subjectLocator { href }
subjectIdentifier = element subjectIdentifier { href }
instanceOf = element instanceOf { ref }

any-markup =
    (text|element * - xtm:* {attribute * {text}*, any-markup*})*
topicMap =   element topicMap
    { version, reifiable, ( topic | association )* }
topic = element topic
    { id, ( itemOrigin | subjectLocator | subjectIdentifier )*,
      instanceOf?, ( topic_name | occurrence )* }
topic_name =  element name
    { reifiable, type?, scope?, text, variant* }
variant = element variant
    { (ref, reifiable, scope?) |
      (reifiable, scope?, text) }
data = element data
    { datatype?, any-markup }
occurrence =  element occurrence
    { ( href, reifiable, type, scope? ) |
      ( datatype?, reifiable, type, scope?, any-markup ) }
association =  element association
    { type, reifiable, scope?, role+ }
role =  element role
    { player, type, reifiable }
```