XTM 2.1 Issues

Leipzig, November 2009

Should we make an XTM 2.1?

- Lars Heuer suggested three changes to XTM 2.0
  - he proposed that we make an XTM 2.1 version based on this
  - his three proposals fit well with the editors' opionions of the weaknesses in XTM 2.0
- However, the biggest issue to be decided is whether to make a new XTM version at all, or to stick with the one we have
- Currently we have 2 XML syntaxes for Topic Maps
  - if we do this we increase the number to 3
  - with all the interchange challenges that this entails
- Does it matter? Is it really a problem?
  - not sure

Current known software

Tool	1.0	2.0	2.1
CmapTools	x		
H-Maps	x		
Isidorus	x	x	
Mappa	x	x	x
Onotoa		x	
Ontopia	x	x	x
Perl TM	x	x	
Ruby Topic Maps	x	x	
TM++	x	x	
TM4J	x		
TMCore	x		
ThinkGraph	x		
Topincs	x	x	
Wandora	x	x	
Wordpress TM	x		
ZTM		x	x?
dtddoc	x		
tinyTiM	x	x	x
tm-converter	x	x	x

Impact of proposed changes

- 2.1 would be backwards compatible with 2.0
  - that is, every 2.0 file would also be a valid 2.1 file *(except for version number?)*
  - every 2.0 file would be imported correctly by a 2.1 reader
  - 2.1 files would not be accepted by 2.0 readers
- Five changes are proposed
  - 1460: don’t require the ID attribute on topic elements
  - 1462: add subjectIdentifierRef and subjectLocatorRef
  - 1459: don’t require fragments in topic references
  - 1493: change ‘version’ attribute value to 2.1 (from 2.0)
  - 1496: additional support for reification

http://projects.topicmapslab.de/projects/xtm/issues

Feedback from vendors

- On issues 1460, 1462, 1459, 1493
  - Robert Cerny: “I would welcome the changes”
  - Arnar Lundesgaard: in favour, “I'm quite certain we will support XTM 2.1 fairly quickly”
- On issue 1496
  - arrived later, so no opinions received yet

1460: don’t require the 'id' attribute on <topic>

- In XTM 2.0 <topic> elements must have an ‘id’ attribute
  - even if they have subject identifiers or subject locators
  - this is stronger than the TMDM requirement
- The proposal is to only require an ‘id’ if there are no other identifiers
  - in other words, one could write simply

```
<topic>
  <subjectIdentifier>http://psi.example.org/foo</subjectIdentifier>
</topic>
```

  - the following would not be allowed

```
<topic><name><value>Foo</value></name></topic>
```

- Benefit
  - not necessary to generate IDs for topics which have other identifiers
    - easier in XSLT conversion, for example
    - this is a real issue, which does occur in real life

1462: add subjectIdentifierRef and subjectLocatorRef

- In XTM 2.0 topics can only be referred to with <topicRef/>
- Benefits (of 2.0)
  - simple to understand and implement
- Downsides (of 2.0)
  - in fragments there must be a topic element for every referenced topic
    - even if the topic is just used as a type or in a scope
  - one cannot refer to a topic only by its PSI
  - causes serious problems for people using large, modular XTM files
    - one major commercial customer had to drop XTM 2.0 because of this

1496: Reference to reifier

- XTM 2.0 supports only the -reifier- attribute, which contains an IDREF
  - this is unlike the other topic references in the syntax
  - suggested solution is to add <reifier> child element, which contains a <*Ref>

```
<association>
   <reifier>
    <subjectIdentifierRef href="http://psi.bouvet.net/lmgs-employment"/>
   </reifier>
   [...]
</association>
```

- The -reifier- attribute may or may not be kept
  - removing it will break backwards compatibility

1459: don’t require fragments in topicrefs

- In XTM 2.0 every <topicRef> URI must contain a fragment
  - <topicRef href=“http://example.org/xtm/foo”/> is illegal
  - <topicRef href=“http://example.org/xtm/#foo”/> is legal
  - <topicRef href=“foo”/> is illegal (and a common error)
  - <topicRef href=“#foo”/> is legal
- Benefits (of 2.0)
  - makes some common errors easy to catch
  - difficult to imagine a case where a non-fragment URI is correct
- Downsides (of 2.0)
  - it’s an odd restriction to have
  - there’s no guarantee that it won’t ever be useful to have a URI without a fragment

1493: change ‘version’ attribute value to 2.1

- In XTM 2.0 files, the version attribute must be exactly “2.0”
- What should the rule be in 2.1?
  1. It must be “2.1” if 2.1 features are used, otherwise it can be either 2.1 or 2.0
  2. It must be “2.1”. 2.0 files are described in the previous version of the spec.
  3. It must be “2.0”. We don’t change the number.
- Cases to consider
  - tool supporting 2.1 and 2.0 gets a 2.0 file
  - tool supporting only 2.0 gets a 2.1 file
  - tool supporting only 2.1 gets a 2.0 file