

CTM - Use Cases

Sam Oh, Sungkyunkwan University, samoh21@gmail.com
Lars Heuer, heuer@semagia.com
Gabriel Hopmans, g.homans@mssm.nl

1. Status of This Document

- This document is an working draft to collect potential use cases and usage scenarios for a Compact Syntax for Topic Maps (CTM).

2. Introduction

- CTM notation would complement the existing XML interchange syntax (XTM), and provide a simple, compact notation for:
 - manually authoring topic maps
 - providing human-readable examples
 - providing a common, lightweight syntax to be used by TMCL and TMQL

3. Use Cases

- Not meant to be complete, but should serve a base for evaluating the existing notations.

More To Do

- Define the more advanced use cases like using namespace directive etc.
- Maybe we should allow more than one solution per use case and per notation, so the notations may show their potential.
- Multiple Subject Identifiers / Locators.
- Unicode!

3.1 Problem Domain

- Assuming that a topic map about the music band *The Beatles* should be created, we define particular aspects of the topic map authoring task.

3.2 Creating Topics

- Contains use cases to create topics.
- Defines use cases for creating type-instance relationships between topics.
- While the type-instance relationship is created with an association, the authors of this document assume that each syntax has a short hand notation for this common use case.

3.2.1. Topic with an Item Identifier

- AsTMa=
 - John
- LTM
 - [John]

• 3.2.2. Typed Topic - Using Item Identifiers

- AsTMa=
 - john (person)
 - Alternative (using the AsTMa= built-in association template "is-a" which creates a type-instance relation between the left and right topic):
 - john is-a person
- LTM
 - [john : person]

• 3.2.3. Typed Topic - Using Subject Identifiers

- AsTMa=
 - Missing (an answer was not provided)
- LTM
 - #PREFIX foo @"http://psi.example.com/"
 - [foo;john : foo:person]

• 3.2.4. Multi Typed Topic - Using Item Identifiers

- AsTMa=
 - john (person musician)
 - or:
 - john is-a person is-a musician
- LTM
 - [john : person musician]
 - fixme - singer, guitarist

• 3.2.5. Multi Typed Topic - Using Subject Identifiers

- AsTMa=
 - john (i'http://psi.example.org/person i'http://psi.example.org/musician)
 - OR:
 - john is-a i'http://psi.example.org/person is-a i'http://psi.example.org/musician
- LTM
 - #PREFIX foo @"http://psi.example.com/"
 - [john : foo:person foo:musician]

3.3. Assigning Topic Names to a Topic

- This section contains use cases to create topic names and their variant names.

• 3.3.1. Topic with an Item Identifier and a Topic Name

- AsTMa=
John
tn: John Lennon
Alternative: Using quotes
John
tn: "John Lennon"
- LTM
[John = "John Lennon"]

• 3.3.2. Topic with a Subject Identifier and a Topic Name

- AsTMa=
ihttp://beatles.com/
tn: The Beatles
- LTM
 - [beatles = "The Beatles" @"http://beatles.com"]
 - fixme: http://psi.beatles.com/The_Beatles/

• 3.3.3. Topic with a Subject Locator and a Topic Name

- AsTMa=
ihttp://beatles.com/
tn: Official website of The Beatles
- LTM
 - [beatles-website = "Official website of The Beatles"
%"http://beatles.com"]

• 3.3.4. Typed Topic Name - Using Item Identifiers

- AsTMa=
john
tn (fullname): John Ono Lennon
- LTM
no support for typed names in LTM 1.3 – might come in 1.4

• 3.3.5. Typed Topic Name - Using Subject Identifiers

- AsTMa=
john
tn (i'http://psi.example.org/fullname): John Ono Lennon
- LTM
ditto

- 3.3.6. Scoped Topic Name - Using Item Identifiers

- AsTMA=
 - john
 - tn @fullname: John Winston Lennon
- LTM
 - [john = "John Winston Lennon" / fullname]

- 3.3.7. Scoped Topic Name - Using Subject Identifiers

- AsTMA=
 - john
 - tn @ihttp://psi.example.org/fullname : John Winston Lennon
- LTM
 - [john = "John Winston Lennon" / foo:fullname]

- 3.3.8. Multi Scoped Topic Name

- AsTMA=
 - beatles
 - tn: The Beatles
 - tn @nickname short: Fab Four
- LTM
 - [beatles = "The Beatles"
= "Fab Four" /nickname short]

- 3.3.9. Typed and Scoped Topic Name

- AsTMA=
 - john
 - tn @yoko (fullname): John Ono Lennon
- LTM
 - no can do (yet)

- 3.3.10. Topic Name with a Variant of datatype String

- AsTMA=
 - john
 - tn: John Lennon
 - var @sortname: lennon, john

#AsTMA= v2.0 assumes automatically that the datatype of the variant is #http://www.w3.org/2001/XMLSchema#string. It is also possible to make #this explicit:

 - john
 - tn: John Lennon
 - var @sortname: "lennon,
 - john^^http://www.w3.org/2001/XMLSchema#string

#It is also possible to shorten this a bit more and define a prefix
#The defined prefix may be used everywhere where a URI is accepted
#Prefix definition

%prefix xs http://www.w3.org/2001/XMLSchema#

#Using the prefix:

 - john
 - tn: John Lennon
 - var @sortname: "lennon, john^^xs:string

- 3.3.10. Topic Name with a Variant of datatype String

- LTM
 - [john = "John Lennon"; "lennon, john"]
- or
 - [john = "John Lennon"
("lennon, john" / iso:sort)]

- 3.3.11. Topic Name with a Variant of datatype XML

- AsTMA=
 - john
 - tn: John Lennon
 - var @markup: "John Lennon^^http://www.w3.org/2001/XMLSchema#anyType
- #With prefix definition
 - %prefix xs http://www.w3.org/2001/XMLSchema#
- john
- tn: John Lennon
- var @markup: "John Lennon^^xs:anyType

- 3.3.11. Topic Name with a Variant of datatype XML

- LTM
 - no can do - XTM 1.1 specific, see LTM 1.4 (or 2.0)

- 3.3.12. Topic Name with a Variant of datatype IRI

- AsTMA=
 - john
 - tn: John Lennon
 - var @image: http://en.wikipedia.org/wiki/Image:Jk_beatles_john.jpg
 - ^^http://www.w3.org/2001/XMLSchema#anyURI
- #With prefix definition
 - %prefix xs <http://www.w3.org/2001/XMLSchema#>
- john
- tn: John Lennon
- var @image: http://en.wikipedia.org/wiki/Image:Jk_beatles_john.jpg^^xs:anyURI

- 3.3.12. Topic Name with a Variant of datatype IRI

- LTM
 - no can do
 - LMG: insufficient desire
 - STP: oversight

- 3.3.13. Topic Name with a Variant of a non-TMDM datatype

- AsTMA=
 - revolution-nine
 - tn: Revolution 9
 - var @number: "9"^^http://www.w3.org/2001/XMLSchema#integer
- LTM
 - no can do - XTM 1.1 specific, see LTM 1.4 (or 2.0)

3.4 Assigning Occurrences to a Topic

- This section defines use cases for creating occurrences.

- 3.4.1. Occurrence of datatype String

- AsTMa=
 - # Note: AsTMa= makes a difference between inline occurrences
 - # (like strings, integers etc.) and external occurrences, that are always URIs
 - # For inline occurrences AsTMa= uses the keyword "in".
 - # For external occurrences AsTMa= v2.0 uses the keyword "ex".
 - # For backward compatibilty it is also possible to qualify an external
 - # occurrence with the keyword "oc"

```
adayinthelife
in: I read the news today, oh boy
```

 - # As with the variants, AsTMa= assumes a string if the datatype
 - # is not qualified. Here the more explicit definition:

```
adayinthelife
in: "I read the news today, oh boy"^^http://www.w3.org/2001/XMLSchema#string
```

 - # Again, using a defined prefix is possible here. Omitted.
 - FIXME: untyped occurrence are not allowed in TMDM. Remove

- 3.4.1. Occurrence of datatype String

- LTM

- 3.4.2. Typed Occurrence of datatype String - Using Item Identifiers

- AsTMa=
 - adayinthelife
 - in (lyrics): I read the news today, oh boy
- LTM
 - {adayinthelife, lyrics, [[I read the news today, oh boy]]}

- 3.4.3. Typed Occurrence of datatype String - Using Subject Identifiers

- AsTMa=
 - adayinthelife
 - in (i http://psi.example.org/lyrics): I read the news today, oh boy
- LTM
 - {adayinthelife, foo:lyrics, [[I read the news today, oh boy]]}

- 3.4.4. Scoped Occurrence of datatype String - Using Item Identifiers

- AsTMa=
 - adayinthelife
 - in @en: I read the news today, oh boy
- LTM
 - {adayinthelife, lyrics, [[I read the news today, oh boy]]} / en

- 3.4.5. Scoped Occurrence of datatype String - Using Subject Identifiers

- AsTMa=
 - adayinthelife
 - in @ihttp://www.topicmaps.org/xtm/1.0/language.xtm#en: I read the news today, oh boy
- LTM
 - {adayinthelife, foo:lyrics, [[I read the news today, oh boy]]} / lang:en

- 3.4.6. Occurrence of datatype XML

- AsTMA=

```
adayinthelife
in: "<html><head><title>A day in the life</title></head><body><p>I read the news today, oh
boy</p></body></html>"^http://www.w3.org/2001/XMLSchema#anyType
```

Alternative: Breaking up the string using triple quotes:

```
adayinthelife
in: '''<html>
<head>
<title>A day in the life</title>
</head>
<body>
<p>I read the news today, oh boy</p>
</body>
</html>'''^http://www.w3.org/2001/XMLSchema#anyType
# Note that this would insert \n into the XML
```

Alternative: Using line continuation with the backslash at
the end of the line. The XML would then be in one line (no \n's inserted)
Using the backslash can be done with simple quoted strings (" or ")
and triple quoted strings ("''' or ''')

```
adayinthelife
in: "<html-\n
<head-\n
<title>A day in the life</title-\n
</head-\n
<body-\n
<p>I read the news today, oh boy</p-\n
</body-\n
</html>"-^http://www.w3.org/2001/XMLSchema#anyType
```

- 3.4.6. Occurrence of datatype XML

- LTM

- no can do (yet), XTM 1.1 specific

- 3.4.7. Occurrence of datatype IRI

- AsTMA=

```
# As said AsTMA= distinguishes between external and internal
# occurrences. External occurrences are automatically datatyped with
# http://www.w3.org/2001/XMLSchema#anyURI
```

```
beatles
ex: http://beatles.com/
```

```
# AsTMA= v1.3 / v2.0 compatibility
beatles
oc: http://beatles.com/
```

```
# This would also be possible but is against our intention
beatles
in: "http://beatles.com/"^http://www.w3.org/2001/XMLSchema#anyURI
```

- 3.4.7. Occurrence of datatype IRI

- LTM

- Bogus use case (untyped occurrence - remove)

- 3.4.8. Occurrence of a non-TMDM datatype

- AsTMA=

```
pennylane
in: "2"^http://www.w3.org/2001/XMLSchema#integer
```

- LTM

- XTM 1.1

3.5 Creating Associations

- An association with a type consisting of a topic with the item identifier "created-by" should be created. A topic with the item identifier "yesterday" should play the role of type "work". Another topic with the item identifier "mccartney" should be the player of the role type "creator".

- 3.5.1. Creating an Association

- AsTMa=
(created-by)
work: yesterday
creator: mccartney
With the definition of an association template the would be:

Association template definition (take care of the [] brackets!)
[(created-by)
work: http://astma.it.bond.edu.au/authoring/psi/1.0#left
creator: http://astma.it.bond.edu.au/authoring/psi/1.0#right
]

The URIs http://astma... are placeholders. The ...#left URI
will be replaced by a topic that occurs at the left side and
the #right URI will be replaced by the topic at the right side

Applying the association template:

yesterday created-by mccartney

The effect of the "yesterday created-by mccartney" statement is
the same as in the first solution

- 3.5.1. Creating an Association

- LTM
 - created-by(yesterday : work, mccartney : creator)

- 3.5.2. Scoped Association

- AsTMa=
@inofficial (created-by)
work: yesterday
creator: mccartney

Using scope is also possible inside an association template:
[@inofficial (created-by)
work: http://astma.it.bond.edu.au/authoring/psi/1.0#left
creator: http://astma.it.bond.edu.au/authoring/psi/1.0#right
]

Applying the association template looks the same as in 3.5.1
yesterday created-by mccartney

- 3.5.2. Scoped Association

- LTM
 - created-by(yesterday : work, mccartney : creator) / unofficial

- 3.5.3. Untyped Association

- AsTMa=
In AsTMa= the empty type (indicated by "()") is
automatically the default type
()
work: yesterday
creator: mccartney
Make it more explicit:
(!http://psi.topicmaps.com/iso13250/association)
work: yesterday
creator: mccartney

- LTM

BOGUS!

- 3.5.4. Untyped Association Role

- AsTMa=
The star * indicates that an "anonymous" default topic
should be created (or if a topic with the subject identifier
http://psi.topicmaps.com/iso13250/association-role already
exists this topic will be used)

(created-by)
*: yesterday
creator: mccartney

- LTM

• BROKEN

- 3.5.5. Supertype-Subtype relationship - Using Item Identifiers

- AsTMA=
 - # AsTMA= has build-in an association template "subclasses" and
 - # the synonym "is-subclass-of" that creates the supertype-subtype
 - # association

```

song subclasses musical-work
song is-subclass-of musical-work
      
```

 - # The two solutions are equal with the explicit association definition:


```

(ihttp://psi.topicmaps.com/iso13250/supertype-subtype )
http://psi.topicmaps.com/iso13250/supertype : musical-work
http://psi.topicmaps.com/iso13250/subtype : song
          
```

- 3.5.5. Supertype-Subtype relationship - Using Item Identifiers

- LTM
 - #PREFIX @"http://www.topicmaps.org/xtm/1.0/core.xtm#"
 - xtm:superclass-subclass(song : xtm:subclass, musical-work : xtm:superclass)

- 3.5.6. Supertype-Subtype relationship - Using Subject Identifiers

- AsTMA=
 - # Using the build-in association template
 - ihttp://psi.example.org/song subclasses ihttp://psi.example.org/musical-work
 - ihttp://psi.example.org/song is-subclass-of ihttp://psi.example.org/musical-work
 - # Equivalent association:


```

(ihttp://psi.topicmaps.com/iso13250/supertype-subtype )
http://psi.topicmaps.com/iso13250/supertype : ihttp://psi.example.org/musical-
work
http://psi.topicmaps.com/iso13250/subtype : ihttp://psi.example.org/song
          
```

- 3.5.6. Supertype-Subtype relationship - Using Subject Identifiers

- LTM
 - #PREFIX @"http://www.topicmaps.org/xtm/1.0/core.xtm#"
 - xtm:superclass-subclass(
 - foo:song : xtm:subclass,
 - foo:musical-work : xtm:superclass)

3.6 Reification

- While the act of reification is well defined inside the (de)serialization process of XTM, the CTM syntax may define another, more explicit reification notation that is conform to the Topic Maps - Data Model (TMDM).
- The following use cases assume that there is no need to use item identifiers / subject identifiers to support reification.

- 3.6.1. Reification of a Topic Map

- AsTMA=
 - # Reification has to be done with a directive


```

%topicmap is-reified-by beatles-topicmap
          
```
 - beatles-topicmap


```

tn: Topic map about The Beatles
          
```
 - # The keyword "is-reified-by" is only used for better readability and
 - # can be omitted. Shorthand:


```

%topicmap beatles-topicmap
          
```
 - beatles-topicmap


```

tn: Topic map about The Beatles
          
```

- 3.6.1. Reification of a Topic Map

```
- LTM
#TOPICMAP -beatles-topicmap
[beatles-topicmap = "Topic map about The Beatles"]
```

- 3.6.2. Reification of a Topic Name

```
- AsTMa=
# Note: You have to put the topic name value into
# quotes to reify a topic name. This has to be done
# to tell the AsTMa= processor where the topic name
# value ends. This is also true for topic variants and
# occurrences
john
tn: "John Ono Lennon" is-reified-by name-of-john-lennon

# DO NOT DO THIS:
# john
# tn: John Ono Lennon is-reified-by name-of-john-lennon

# Result: Value of the topic name is
# "John Ono Lennon is-reified-by name-of-john-lennon"
# As shorthand "reified-by" can also be used:
john
tn: "John Ono Lennon" reified-by name-of-john-lennon
```

- 3.6.2. Reification of a Topic Name

```
- LTM
[john = "John Ono Lennon" ~name-of-john-lennon]
```

- 3.6.3. Reification of a Variant

```
- AsTMa=
john
tn: John Lennon
var @sortname: "lennon, john" is-reified-by johns-sortname
# Explicit datatype:
john
tn: John Lennon
var @sortname: "lennon,
john"~"http://www.w3.org/2001/XMLSchema#string is-reified-by johns-
sortname

- LTM
[john = "John Lennon"
("lennon, john" / sortname ~ johns-sortname) ]
```

- 3.6.4. Reification of an Occurrence

```
- AsTMa=
# For external occurrences no quotes are necessary
# because a space indicates the end of a URI
john
ex: http://johnlennon.com/ reified-by lennons-website

lennons-website
tn: Official website of John Lennon

- LTM
{john, website, "http://johnlennon.com/" } -lennons-website
[lennons-website = "Official website of John Lennon"]
```

- 3.6.5. Reification of an Association

```
- AsTMa=
(partnership) is-reified-by lennon-mccartney
partner: lennon
partner: mccartney

lennon-mccartney
tn: Lennon / McCartney

- LTM
partnership(lennon : partner, mccartney : partner) -lennon-mccartney
[lennon-mccartney = "Lennon / McCartney"]
```

- 3.6.6. Reification of an Association Role

- AsTMa=
 - Is not possible with AsTMa=
- LTM
 - partnership(
 - lennon : partner -lennon-qua-partner-of-paul,
 - mccartney : partner)
 - [lennon-qua-partner-of-paul = "Lennon, the partner of McCartney"]

3.7 Comments

- The CTM should support comments to annotate the topic map content.

- 3.7.1. Singleline Comment

- AsTMa=
 - This is a topic map about The Beatle
- LTM
 - /* ...and it sucks */

- 3.7.2. Multiline Comment

- AsTMa=
 - # This is a topic map about The Beatles
 - # Authored by John Doe
- LTM
 - /*
 - he
 - should
 - be
 - shot @
 - */

Additional use cases

- nested assertions
 - describe one object inside another
 - created-by (
 - [yesterday = "Yesterday"] : work,
 - [mccartney = "McCartney"] : creator)
 - Get example from Dmitry in PTM
 - ref. RDF N3 notation
- simplified binary relationships
 - even more simplified hierarchical relationships
- ordered values*
 - e.g. order of authors of a book
- arbitrary character encodings
- escape syntax for special characters

more use cases...

- support for metadata on statements, etc.*