TMDM-TMRM mapping

October 13, 2007

Status

- **One mapping was defined at the Seoul meeting in May 2006**
- **I wrote this up properly in my July 2, 2007 draft**
- **Robert then edited and incorporated a mapping in the latest TMRM draft (N0886)**
- ***However,* the mapping in the latest TMRM draft is very different!**
  - the presentation is different (and much superior)
  - the mapping itself is also different (and not necessarily superior)
- **We discussed this in Montréal in August, but did not really get very far**
- **Proposal**
  - we go through both mappings in paralell here, and review them

Datatypes

- **A datatype consists of**
  - an identifying IRI (as per TMDM)
  - a set of strings (lexical space)
  - a set of values (value space)
  - a string-to-value function
  - a value-to-string function
  - a total ordering on the value set

- **A datatype consists of**
  - an identifying IRI (as per TMDM)
  - a set of strings (lexical space)
  - a set of values (value space)
  - a string-to-value function
  - a value-to-string function

We include the ordering in datatypes

Type and subtype

- **Special proxies are used for typing and subtyping**
- **These proxies are *not* associations**
  - { <type, [typing topic]>, <instance, [instance proxy]> }
- **The same proxies are used for type-instance and for the [type] property**
- **Type and subtype are represented like all other associations**
- **The [type] property is just a key in proxies representing statements**

We let type and subtype look the same as other assocs. Remove these proxies from B.2, but keep definition of sub/isa relations.

Ontological commitments

- **One mapping generates two subject maps**
  - one containing exactly what was in the original topic map,
  - and one containing all implied information
- **One mapping generates a single subject map**
  - containing both

a isa b
b ako c
a isa c

Issue: how do we know that there should be three *associations* here?

Requirement:
- must be possible to query with and without inferred info,
- must be possible to see from the query text which of these is the case for any given query.

Proposed solution: use the import declarations in TMQL to import either an environment with inferencing or an environment without. These two environments are standardized in TMQL, but it's possible to import other non-standardized environments. The environment *with* inferencing is the default environment.

The topic map proxy

- **{<topic, ____>+, <association, ____>+}**
- **tmproxy isa topicmap**

- **{<type, topicmap>}**

We prefer the one on the right

Topic proxies

- **{<item-identifier, i>+,**
  <subject-identifier, i>+,
  <subject-locator, i>+}

- **{<item-identifier, i>+,**
  <subject-identifier, i>+,
  <subject-locator, i>+}

Topic name proxies

- **{<scope, [scope proxy]>, <subject, [topic]>, <value, “....”>}**
- **topicnameproxy isa [type]**

- **{<scope, [scope proxy]>, <subject, [topic]>, <value, “....”>, <type, [type]>}**

We go with the right-hand one

The other proxies

- They follow the same pattern
- One proposal has type inside the proxy, the other has them outside

Constraints

- **One proposal has a set of constraints that valid TMDM instances represented as subject maps must conform to**
- **The other proposal leaves this to be implied by the TMDM->TMRM mapping**
- **The question is, do we think the constraints are necessary?**

We want the constraints

Axes

- **Both proposals contain definitions of the TMQL axes as mathematical relations**
  - the definitions are necessarily different because the representations are different
  - no point in dicussing this here

Proposal A

```
#PREFIX tm @"http://bogus.garshol.priv.no/tmdm/"
#PREFIX tmdm @"http://psi.topicmaps.org/iso13250/model/"

[tm:thing]
  [tm:subject]
  [tm:statement]
    [tm:association]
    [tm:characteristic]
      [tm:occurrence]
      [tm:name]

[tmdm:type-instance]
  [tmdm:type]
  [tmdm:instance]

[tmdm:supertype-subtype]
  [tmdm:supertype]
  [tmdm:subtype]

tmdm:supertype-subtype(tm:thing : tmdm:super, tm:subject : tmdm:sub)
tmdm:supertype-subtype(tm:thing : tmdm:super, tm:statement : tmdm:sub)
tmdm:supertype-subtype(tm:statement : tmdm:super, tm:association : tmdm:sub)
tmdm:supertype-subtype(tm:statement : tmdm:super, tm:characteristic : tmdm:sub)
tmdm:supertype-subtype(tm:characteristic : tmdm:super, tm:occurrence : tmdm:sub)
tmdm:supertype-subtype(tm:characteristic : tmdm:super, tm:name : tmdm:sub)
```

Ditch this one, use B.

Proposal B

```
%prefix tm   http://psi.topicmaps.org/iso13250/model.2/
%prefix tmdm http://psi.topicmaps.org/iso13250/model/

subject            tm:subject .
topic              tmdm:topic        iko tm:subject .
statement          tmdm:statement    iko tm:subject .
association        tmdm:association  iko tmdm:statement .
characteristic     tm:characteristic iko tmdm:statement .
topic-name         tmdm:topic-name   iko tm:characteristic .
occurrence         tmdm:occurrence   iko tm:characteristic .

member             tm:member . # for set membership
instance           tmdm:instance .
type               tmdm:type .
subtype            tmdm:subtype .
supertype          tmdm:supertype .

topicmap           tmdm:topicmap .
item-identifier    tmdm:item-identifier .
subject-identifier tmdm:subject-identifier .
subject-locator    tmdm:subject-locator .
scope              tmdm:scope .
value              tm:value .
```

- Replace topic-name with name
- Add topicmap under subject
- Add variant under statement
- Update to latest CTM

Scope issues!

- **LMG scope presentation from TMRA 2007**
  - issue of whether statements imply other statements with narrower scopes (slide 18)
  - issue of how inferencing interacts with scope (slide 19)
- **We try to solve this by**
  - introducing scope operators which honour these semantics
  - put these either in the mapping

Subtyping of statements

- **composed-by(composer : puccini, composition : la-wally)**
- **composed-by ako created-by**
- **composer ako creator**
- **composition ako creation**
- ***created-by(creator : puccini, creation : la-wally)***
- ***created-by(composer : puccini, creation : la-w***
- ***created-by(creator : puccini, composition : la-wally)***
- ***created-by(composer : puccini, composition : la-wally)***
- ***composed-by(composer : puccini, creation : la-wally)***
- ***composed-by(creator : puccini, composition : la-wally)***
- ***composed-by(composer : puccini, composition : la-wally)***

Not clear if the red associations are implied by the black ones. At the top of the type hierarchy the associations will become symmetrical...
Not entirely sure how to deal with this in the mapping and in TMCL.
Need to find requirements for these situations. LM will find the reqs. (See slide 25.)

Names, occurrences, and associations

- **TMDM says**
  - occurrences are a kind of association, and
  - names are a kind of occurrence
- **For example,**
  - topic occtype: http://example.org .
  - *occtype(resource : rtopic, subject : topic)*
  - *rtopic = http://example.org .*
- **The question is: should this be formalized?**
  - no, we don't think so

Feedback

The sub and isa relations

- **TMRM defines two classes of relations: isa and sub**
- **They should be called isa and ako, to be consistent with CTM**

Formal semantics

- **Annex B will have the title “Formal semantics of TMDM”**
  - this will describe the formal semantics/ontological commitments of the TMDM

Separation of kinds of implication

- **There is a general feeling that the separation between explicit and implied information is insufficient, and that a finer-grained separation is necessary**
- **For example between**
  - explicit proxies,
    - puccini isa composer
  - implied-and-invariant proxies,
    - topic type isa subject
  - implied-and-variant-but-ontology-independent proxies, and
    - puccini isa subject
  - implied-and-variant-and-ontology-dependent proxies
    - puccini isa person (implied by explicit subtyping)

Environments are defined for the first, and the union of the last three.

- **This does not need to be represented with separate subject maps, but can be done by annotating the proxies in a single map**

Need for roundtripping

- **Need to specify in addition to the TMDM-TMRM mapping**
  - an inverse mapping from TMRM to TMDM, *and*
  - the constraints on the TMRM instances where this will actually work
- **This implies that the constraint part that is absent from *one* of the mapping proposals is actually needed**

Reified and unreified subjects

- **Topics reifying other information items in the source TMDM need to be mapped to a *single* proxy in the TMRM which is produced from *both* the topic and the information item, and which represents the subject that they collectively represent**

This is logically/conceptually correct.
However, it creates problems for TMQL (the reifier axis) and it also breaks the structure of TMDM subject maps.
We might rescue this with a bowtie merging rule that implements this.

A more general mapping

- **The current mapping achieves the purpose of supporting TMQL, but has some limitations**
  - for example, it does not allow information from different topic maps to be merged into a single subject map in such a way that it is apparent what the source of each piece of information is
  - there is a feeling that there is also a need for a mapping that supports this
  - such a mapping should be produced at some later stage, and has been added to the list of work items to be started in the future

Requirements

- **We need a document that states what the purpose of (and requirements for)**
  - the TMDM->TMRM mapping and
  - the formal semantics
- **are**
- **This should be published as a separate N0xxx document in the ISO repository**
  - Lars Marius is going to write this, and Jaeho Lee is going to criticize it
  - see next slide

Summary

- **We are ready to finish the TMDM -> TMRM mapping**
  - LMG will do this before November 12
  - Robert will put the current TMRM into cvs.garshol.priv.no
- **We are not ready to finish the inferencing part**
  - the requirements will be written by LMG before November 12
  - we *can* write up a strawman proposal
    - LMG will do this before November 12
- **This is urgent!**
  - TMCL and TMQL have to wait for this...