TMDM mapping to TMRM

Lars Marius Garshol
Bouvet
<larsga@bouvet.no>

July 2, 2007

1 Introduction

This document describes a proposed representation of TMDM in TMRM. The intention is that this will be reviewed by SC34, and if approved included in ISO 13250-5 as an annex. The document is couched in a style known to be inappropriate for direct inclusion, but, it is hoped, suitable as a basis for discussion.

This mapping is based on document N710 in the SC34 document repository. Owing to lack of time this document is still an incomplete draft.

The mapping from TMDM to TMRM consists of three parts:

- The actual transformation from TMDM to TMRM.
- The ontological commitments.
- The constraints on the model.

1.1 General principles

Two general principles have been followed in the design of this mapping:

- The properties of a proxy should be as exact a representation of its identity as possible. Information about the subject which does not contribute to its identity should be external to the proxy representing the subject. (A related principle is that proxies should not contain artificial identifiers whose only purpose is to make proxies distinguishable.)
- The proxies representing different things in the TMDM model should be as uniform in structure as possible, in order to make the representation easier to understand and follow.

The entire mapping follows from these two principles.

1.2 Readable overview

This section presents what the author claims to be an intelligible overview of the mapping. It uses normal set and tuple syntax to represent proxies and properties. It also uses `+` to represent required and repeatable properties, `*` for optional and repeatable properties, and `?` for optional properties.

The TMRM representation of TMDM primarily consists of proxies representing either topics or statements. (Statements in TMDM are associations, occurrences, topic names, and variants.)

1.2.1 Statements

The proxies for statements have the following form:

```
{(type,      ____),
 (scope,     ____),
 (roletype1, ____),
 (roletype2, ____),
 ...}
```

Of course, topic names, occurrences, and variants don't have role types in TMDM, but in this representation they have been extended so that they do.

The statements for occurrences take the following form:

```
{(type,     ____),
 (scope,    ____),
 (subject,  ____),
 (resource, ____)}
```

The value of `subject` is here the topic the occurrence is attached to, and `resource` is the actual occurrence.

The statements for topic names take the following form:

```
{(type,    ____),
 (scope,   ____),
 (subject, ____),
 (value,   ____)}
```

Again, the `subject` is the topic, and `value` is the actual name.

The statements for variant names take the following form:

```
{(type,    variant),
 (scope,   ____),
 (subject, ____),
 (value,   ____)}
```

Here the `subject` is the topic name the variant is attached to. The type is hard-wired because variant names do not have any type, and we need to identify this somehow as being a variant.

1.2.2 Topics

The topic proxies have the following form:

```
{(itemid,  _____)*,
 (subjid,  _____)*,
 (subjloc, _____)*,
 (reifies, _____)}
```

In other words: topic proxies only contain the identity of the topic, and nothing else.

1.2.3 The rest

This is of course not everything. There are two more kinds of proxies: scope proxies and the topic map proxy. The scope proxies are the values of the scope properties in the statements, and they have the following form:

```
{(theme, ____)*}
```

The topic map proxy is the proxy that represents the topic map item in TMDM. There is only ever one of these, and the only thing it is used for is when a topic reifies the topic map. It takes the following form (which ensures that there can only be one in each topic map):

```
{(type, topicmap)}
```

2 Datatypes

A datatype is a 5-tuple $(S, V, t, t^{-1}, <)$ as follows:

- S is a set of strings. This is the lexical space of the datatype; that is, the set of strings which are valid textual encodings of values belonging to the datatype.
- V is a set of values (that is, $V \subset \mathcal{V}$). V is the set of all values belonging to the datatype.
- t is a function $S \mapsto V$.
- t^{-1} is a function $V \mapsto S$. Further, for all $v \in V$ it must hold that $t(t^{-1}(v)) = v$.

- $<$ is a total ordering on V.

The set $\mathcal{D}$ is the set of all (u, d) pairs where d is a datatype and u is a PSI identifying that datatype.

3 The mapping

The mapping takes the form of preconditions and postconditions. In order to make the mapping more readable we need a notation for accessing individual parts of the TMDM model. In the notation used here if i is an information item, then $i.property$ refers to the value of the property of i whose name is `[property]`

The mapping itself is performed by the $m_m(m)$ function, which takes a TMDM topic map item as input and produces the corresponding TMRM map as output.

NOTE: Reification is not handled.

3.1 Predefined proxies

The following predefined proxies are used by the TMDM mapping. P is the set of all proxies used by the mapping.

$$
\begin{aligned}
\texttt{twisted} &= \{(twisted, twisted)\} \\
\texttt{type} &= \{(twisted, \text{"}type\text{"})\} \\
\texttt{topicmap} &= \{(twisted, \text{"}topicmap\text{"})\} \\
\texttt{itemid} &= \{(twisted, \text{"}itemid\text{"})\} \\
\texttt{subjid} &= \{(twisted, \text{"}subjid\text{"})\} \\
\texttt{subjloc} &= \{(twisted, \text{"}subjloc\text{"})\} \\
\texttt{scope} &= \{(twisted, \text{"}scope\text{"})\} \\
\texttt{value} &= \{(twisted, \text{"}value\text{"})\} \\
\texttt{subject} &= \{(twisted, \text{"}subject\text{"})\} \\
\texttt{resource} &= \{(twisted, \text{"}resource\text{"})\} \\
\texttt{variant} &= \{(twisted, \text{"}variant\text{"})\} \\
\texttt{theme} &= \{(twisted, \text{"}theme\text{"})\}
\end{aligned}
$$

3.2 The topic map

A TMRM map for a TMDM topic map is produced as follows:

$$
\begin{aligned}
m_m(m) = \quad & \{(\texttt{type}, \texttt{topicmap})\} \cup \\
& \{m_t(t) | t \in m.topics\} \cup \\
& s(m) \cup \\
& \{s | (\texttt{scope}, s) \in t \wedge t \in s(m)\} \cup \\
& P
\end{aligned}
$$

The s function is defined as follows:

$$
\begin{aligned}
s(m) = \quad & \{m_a(a) | a \in m.associations\} \cup \\
& \{m_n(n) | n \in m.topics.topic\_names\} \cup \\
& \{m_v(v) | v \in m.topics.topic\_names.variants\} \cup \\
& \{m_o(o) | o \in m.topics.occurrences\}
\end{aligned}
$$

3.3 Topic items

Topic items are converted as follows:

$$
\begin{aligned}
m_t(t) = \quad & \{(\texttt{itemid}, u) | u \in t.item\_identifier\} \cup \\
& \{(\texttt{subjid}, u) | u \in t.subject\_identifier\} \cup \\
& \{(\texttt{subjloc}, u) | u \in t.subject\_locator\}
\end{aligned}
$$

3.4 Topic name items

Topic name items are converted as follows:

$$
\begin{aligned}
m_n(n) = \quad & \{(\texttt{type}, m_t(n.type)), \\
& (\texttt{scope}, m_s(n.scope)), \\
& (\texttt{subject}, m_t(n.parent)), \\
& (\texttt{value}, n.value)\}
\end{aligned}
$$

3.5 Variant name items

Variant names are converted as follows:

$$
\begin{aligned}
m_v(v) = \quad & \{(\texttt{type}, \texttt{variant}), \\
& (\texttt{scope}, m_s(v.scope)), \\
& (\texttt{subject}, m_n(v.parent)), \\
& (\texttt{value}, t_v(v.datatype, v.value))\}
\end{aligned}
$$

Datatype!

3.6 Occurrence items

Occurrences are converted as follows:

$$
\begin{aligned}
m_o(o) = \quad & \{(\texttt{type}, m_t(o.type), \\
& (\texttt{scope}, m_s(o.scope)), \\
& (\texttt{subject}, m_t(o.parent)), \\
& (\texttt{resource}, t_v(o.datatype, o.value))\}
\end{aligned}
$$

Datatype!

3.7 Association items

Associations are converted as follows:

$$
\begin{aligned}
m_a(a) = \quad & \{(\texttt{type}, m_t(a.type), \\
& (\texttt{scope}, m_s(a.scope))\} \cup \\
& \{(t, p) | \exists r \in a.roles : r.type = t \wedge r.player = p\}
\end{aligned}
$$

3.8 Scope

Scopes are converted as follows:

$m_s(s) = \{(\texttt{theme}, t) | t \in s\}$

3.9 Datatype translation

Values are translated as follows:

$t_v(u, v) = t_t(u) \circ (v)$

where $t_t(u)$ is:

$t_t(u) = t | (u, (S, V, t, t^{-1}, <)) \in \mathcal{D}$

4 Constraints

The $\odot$ binary operator is defined as:

$S \odot S' = \{(u, v) | u \in S \wedge v \in S'\}$

In the following, U refers to the value set of all URIs, and S' to the value set of all strings.

The following sets of proxies are the sets of all proxies representing various TMDM constructs in TMRM maps:

$$
\begin{aligned}
T = \quad & 2^{\{\texttt{itemid},\texttt{subjid},\texttt{subjloc}\} \odot U} - \emptyset \\
N = \quad & \{n | n = \{t, s, p, v\} \wedge \\
& t \in \{\texttt{type}\} \odot T \wedge \\
& s \in \{\texttt{scope}\} \odot S \wedge \\
& p \in \{\texttt{subject}\} \odot T \wedge \\
& v \in \{\texttt{value}\} \odot S'\} \\
V = \quad & \{v | v = \{(\texttt{type}), \texttt{variant}), s, p, v\} \wedge \\
& s \in (\{\texttt{scope}\} \odot S) - \emptyset \wedge \\
& p \in \{\texttt{subject}\} \odot N \wedge \\
& v \in \{\texttt{value}\} \odot \mathcal{V}\} \\
O = \quad & \{o | o = \{t, s, p, v\} \wedge \\
& t \in \{\texttt{type}\} \odot T \wedge \\
& s \in \{\texttt{scope}\} \odot S \wedge \\
& p \in \{\texttt{subject}\} \odot T \wedge \\
& v \in \{\texttt{value}\} \odot \mathcal{V}\} \\
A = \quad & \{a | \{t, s\} \cup r \wedge \\
& t \in \{\texttt{type}\} \odot T \wedge \\
& s \in \{\texttt{scope}\} \odot S \wedge \\
& r \subset T \odot T \wedge \\
& r \neq \emptyset\} \\
S = \quad & \{\texttt{theme}\} \odot T \\
M = \quad & \{\{(\texttt{type}, \texttt{topicmap})\}\}
\end{aligned}
$$

For a TMRM map m to be a valid representation of a TMDM topic map, the following constraints must be met.

$m \subset (T \cup N \cup V \cup O \cup A \cup S \cup M \cup P)$
$P \subset m$
$M \subset m$
$\nexists t, t' \in m | t \in T \wedge t' \in T \wedge t \cap t' \neq \emptyset$

5 Ontological commitments

Some statements in topic maps imply the validity of other statements. This section defines a mapping from any TMRM topic map to a new TMRM topic map containing the implied statements.

5.1 Static part

The following is present in all such topic maps (in LTM syntax). Note that the URI of the `tm` PSI namespace is not yet defined.

```
#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)
```

ISSUE: Should names for these topics be included?

The following proxy labels are used in the rest of this section. Note that the `tmdm` prefix is used for brevity.

$$
\begin{array}{ll}
\texttt{subject'} & = \{(subjid, \texttt{tm:subject})\} \\
\texttt{instance-of} & = \{(subjid, \texttt{tmdm:type-instance})\} \\
\texttt{type'} & = \{(subjid, \texttt{tmdm:type})\} \\
\texttt{instance} & = \{(subjid, \texttt{tmdm:instance})\} \\
\texttt{super-sub} & = \{(subjid, \texttt{tmdm:supertype-subtype})\} \\
\texttt{super} & = \{(subjid, \texttt{tmdm:supertype})\} \\
\texttt{sub} & = \{(subjid, \texttt{tmdm:subtype})\}
\end{array}
$$

5.2 Semantics of subject

Every topic is an instance of `tm:subject`, as follows:

$$
\begin{array}{lll}
p_1 = \{a | a = & \{(\texttt{type}, \texttt{instance-of}), \\
& (\texttt{scope}, \emptyset), \\
& (\texttt{type'}, \texttt{subject'}), \\
& (\texttt{instance}, t)\} \wedge \\
& t \in m \wedge \\
& t \in T\}
\end{array}
$$

5.3 Subtyping transitivity

The `tmdm:supertype-subtype` association type is transitive, and this implies new relationships of the same type. The $tsubd_m t'$ relation is defined as follows:

$tsubd_m t' \Leftrightarrow \{(\texttt{type}, \texttt{super-sub}), (\texttt{scope}, s), (\texttt{super}, t), (\texttt{sub}, t')\} \in m$

ISSUE: This ignores scope! That's not good! We'll need to consider what to do about that.

The $tsub_m t'$ relation is then defined as:

$tsub_m t' \Leftrightarrow tsubd_m t' \vee (tsubd_m t'' \wedge t'' sub_m t')$

The following associations are added to the generated map:

$p_2 = \{a | a = \{(\texttt{type}, \texttt{super-sub}), (\texttt{scope}, \emptyset), (\texttt{super}, t), (\texttt{sub}, t')\} \wedge tsub_m t'\}$

5.4 Instance-of semantics

The $iisad_m t$ relationship is defined as follows:

$tisad_m t' \Leftrightarrow \{(\texttt{type}, \texttt{instance-of}), (\texttt{scope}, s), (\texttt{type'}, t), (\texttt{instance}, i)\} \in m$

The $iisa_m t$ relation is then defined as:

$iisa_m t \Leftrightarrow iisad_m t \vee (iisad_m t' \wedge t' sub_m t)$

The following associations are added to the generated map:

$p_3 = \{a | a = \{(\texttt{type}, \texttt{instance-of}), (\texttt{scope}, \emptyset), (\texttt{type'}, t), (\texttt{instance}, i)\} \wedge iisa_m t\}$

5.5 Semantics of statement

This part is not ready yet. The same goes for "name", "occurrence", and "association".

5.6 Semantics of characteristic

6 Axes

This section is going into the TMQL standard, but has been written here for the convenience of the author.

$i\ types_m\ t$	$\Leftrightarrow i\ isa_m\ t$
$t\ supertypes_m\ t'$	$\Leftrightarrow t\ sub_m\ t'$
$a\ players_m\ t$	$\{(a,t)\|\exists r,p:(r,t)\in p\wedge p\in m\wedge r\notin\{verb@type@,\texttt{scope}\}\}$
$a\ roles_m\ t$	$\{(a,t)\|\exists t',p:(t,t')\in p\wedge p\in m\wedge t\notin\{verb@type@,\texttt{scope}\}\}$
$t\ characteristics\ c$	$\Leftrightarrow c\rightarrow\texttt{subject'}=\{t\}\wedge(c\rightarrow\texttt{type})sub_m\texttt{tm:characteristic}$
$s\ scope\ t$	$\{(s,t)\|\exists p:(\texttt{scope},p)\in s\wedge(\texttt{theme},t)\in p\}$
$t\ locators\ u$	$\{(t,u)\|(\texttt{subjloc},u)\in t\}$
$t\ indicators\ u$	$\{(t,u)\|(\texttt{subjid},u)\in t\}$
$reifier$	???

The characteristic relation assumes information that is only in the generated semantics map. How to add?