TMCL Issues

Leipzig, November 2009

Current status of TMCL

- We have a draft that's been through a few iterations
  - we think it's fairly solid
  - however, it's not yet implemented, as far as we know
- The standard is currently at FCD
  - next ballot is FDIS, which means *no more changes of substance*
- So we really need to get a solid draft together
  - *then* go to ballot
- We've spent a lot of time on this
  - we really need to wrap this one up and move on

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

1458: Referring to validation rules

- Feedback from Seattle meeting:
  - "Global validation rules section numbering. We need it for referring to them.”
- However
  - this would bloat the table of contents dramatically
  - it would also inflate the size of the text by introducing lots of new headings
  - it gets worse because text before rule needs a separate subclause
  - *there are no clauses which have more than one global validation rule*
- Proposal
  - just say “clause 7.5 global validation rule”, or
  - “7.5 GVR”

1447: Error messages for user-defined constraints

- How are violations of user-defined constraints to be reported?
  - current spec doesn’t say anything about it
  - a possible convention would be to use at least the name of the constraint topic in the report
  - however, not clear if TMCL should say anything about this
- Alternatively, one could use TMQL expressions to build the error message

```
c isa tmcl:denial-constraint;
  - "Person died before being born";
  tmcl:validation-expression: ” // person [ . / date-of-birth <= . / date-of-death ]  ";
  tmcl:error-message: "     'Person ' + $this + ' died ' + $this / date-of-death + ', but
     was born ' + $this / date-of-birth  " .
```

1474: Item identifier constraints

- Should there be a constraint type for this?
- Arguments against
  - inflates the size of the spec
  - not clear that there is much need
  - can be solved using user-defined constraints
- Arguments for
  - looks odd to have them for subject-* and not item-identifiers

1490: Datatype for regular expressions?

- Should there be one?
  - the tmcl:regexp occurrence type would then use it
- Arguments against
  - requires TMCL to define one extra datatype,
  - requires processors to support one extra datatype,
  - XML Schema (which defined the language) has no datatype for it
- Arguments for
  - ?

How are TMCL schemas identified?

- Using subject locators?

```
tmcl-schema isa tmcl:schema;
  = http://www.isotopicmaps.org/tmcl/2009-10-26/schema.ctm ;  # location of file
  - "TMCL schema” .
```

- Or using subject locators?

```
tmcl-schema isa tmcl:schema;
  http://psi.topicmaps.org/tmcl/meta-schema/2009-10-26 ;  # PSI
  - "TMCL schema” .
```

1446: Indicating the schema used

- Should topic maps have a way to say what schema they follow?
  - general agreement that if so, this is just documentation
  - that is, the user decides what schema to validate against, not the topic map
- Some possible approaches
  1. An occurrence on the reified topic map refers to the schema file.
  2. An association from the reified topic map connects to a schema topic.
  3. Typing topics connect with a schema topic via tmcl:is-defined-by.

1473: tmcl:imports

- Should there be a way to import another TMCL schema?
  - at the syntactic level there already is (%include / <mergeMap>)
  - however, at the logical level there is no support for this
- Similar proposals to 1446
  - occurrence type referring to imported file
  - associations to schema topic representing schema to be imported
- What are the semantics?
  - validate against this schema, too, but don’t mess up my current topic map?
  - same as mergemap?

1495: min/max values for occurrences

- Should it be possible to constrain the range of values for occurrences?
  - for example: the value of the “age” occurrence must be in the range 0-150
- Possible resolutions:
  1. Add a new occurrence-value-range-constraint with min/max occurrences.
  2. Don’t do anything, because this can be done with user-defined constraints.
  3. Consider value restrictions to be part of the datatype definition. (XSD does.)
     - unfortunately, TMCL does not provide support for user-defined datatypes

847: Variant constraints

- Rejected in Seattle, but users not happy
- Proposal:

```
def has-variant($tt, $nt, $t, $min, $max)
  ?c isa tmcl:variant-constraint;
    tmcl:card-min: $min;
    tmcl:card-max: $max.

  tmcl:constrained-topic-type(tmcl:constrains : ?c, tmcl:constrained : $tt)
  tmcl:constrained-statement(tmcl:constrains : ?c, tmcl:constrained : $nt)
  tmcl:required-scope(tmcl:constrains : ?c, tmcl:constrained : $t)
end
```

1505: Simplify TMCL structure

I'd like to propose to simplify the internal structure of the TMCL schema (without changing its behavior). That means to represent all constraints not as topic-types but as association-types and respectivily all association-types the constraints are involved with, as the corresponding role-types. This is possible because a) all constraints have a connection to the constrained topic-type through exactly one association b) all these associations are binary 1-1-associations (the min and max cardinality equals 1 for both roles).So a) fulfils the association pattern an b) the role pattern.

- Some constraints hold additional data like min max cardinalities in occurrences. I suggest to represent that in a topic (having the same occurrences as the constraint before) which either a) reifies the constraint association or b) plays an additional role in it.

1506: Renaming of role constraints

- Michael Quaas suggests renaming
  - topic-role-constraint to plays-role-constraint
  - association-role-constraint to has-role-constraint

1507: role-combination-constraint for n-ary

- The constraint already supports n-ary associations
  - but only allows specifying rules for pairs of roles
- Michael Quaas suggests extending it
  - not clear how to represent arbitrarily large rules

1521: constrained-association

I find it more constistent when for example the plays-role template would look like this:

tmcl:constrained-topic-type(tmcl:constrains : ?c, tmcl:constrained : $tt)

tmcl:constrained-ASSOCIATION(tmcl:constrains : ?c, tmcl:constrained : $at)

tmcl:constrained-role(tmcl:constrains : ?c, tmcl:constrained : $rt)

instead of

tmcl:constrained-topic-type(tmcl:constrains : ?c, tmcl:constrained : $tt)

tmcl:constrained-STATEMENT(tmcl:constrains : ?c, tmcl:constrained : $at)

tmcl:constrained-role(tmcl:constrains : ?c, tmcl:constrained : $rt)

1606: Multiple association signatures

- The current draft does not allow specifying association types like
  - works-for(person, company)
  - works-for(person, non-profit-organization)
- The editors don’t believe that this is good modelling
  - either there should only be a single signature, or
  - if the difference is meaningful, two separate association types