TMQL 0.1 – Design Notes

Notes from WG3 meeting Amsterdam, April 14-15 2004 regarding ISO 18048: Topic Maps Query Language

Steve Pepper, Convenor, SC34/WG3, <pepper@ontopia.net>
How these slides were prepared

1. Four languages for querying topic maps were presented:
   1. AsTMa?
   2. TMPath
   3. tolog
   4. Toma

2. The strengths and weaknesses of each were discussed and noted, and consensus reached on general issues of style and structure for TMQL.

3. A draft synthesis was prepared by the group, by taking one of the candidates that was already quite close to this style (tolog) and noting the (non-trivial) changes required. These are described in the slides that follow.

4. Finally, outstanding issues were noted & a plan of work outlined - see end of slides.
SELECT $opera
WHERE composed-by( $opera : opera, puccini : composer )
?
Allow role types to be (sometimes?) omitted

SELECT $opera WHERE
WHERE
  composed-by( $opera, puccini )
?

SELECT $opera WHERE
WHERE
  composed-by( $opera, $composer )
?
Add variables for associations

SELECT $puccini-operas
WHERE
  composed-by( $opera : opera, puccini : composer )
  AS $puccini-operas
?
Add path-based capabilities

SELECT $opera/@bn/english
WHERE
  composed-by( $opera : opera, puccini : composer )
ORDER BY $opera/premiere-date
?

SELECT $opera/@bn/english
WHERE
  composed-by( $opera, puccini ),
  $opera/premiere-date < "1900-01-01"
?

http://www.isotopicmaps.org
Add functions

Both built-in...

SELECT upper( $opera/@bn/english )
WHERE
  composed-by( $opera : opera, puccini : composer)
ORDER BY $opera/premiere-date
?
Add functions

...and user-defined

```
IMPORT "http://.../.../" AS foo

sounds-like( $N, $M) :-
    foo:soundex($N) == foo:soundex($M).

SELECT $opera/@bn/english
WHERE
    composed-by( $opera : opera, $composer : composer),
    sounds-like( $opera/@bn/english, "toska" )
ORDER BY $opera/premiere-date
?```
Simplify instance-of()

SELECT $city, COUNT($opera)
WHERE
  $city : city,
  { premiere($opera, $city) | 
    premiere($opera, $theatre),
    located-in($theatre, $city) } 
ORDER BY $opera DESC?
Add FLOWR constructors

FOR $city
WHERE
$cite : city,
{ premiere($opera, $city) |
    premiere($opera, $theatre),
    located-in($theatre, $city) }
ORDER BY $opera DESC?
RETURN
$city no="COUNT($opera)">$city</city>
Query multiple topic maps

Needs to be possible. Requires further elaboration.
Some ideas:

```
SELECT $opera
FROM opera.xtm, wagner.xtm, janacek.ltm
WHERE
    composed-by( $opera : opera, $composer : composer )?

FOR $opera IN opera.xtm, wagner.xtm, janacek.ltm
WHERE
    composed-by( $opera : opera, $composer : composer )?

FOR $opera IN opera.xtm FOR $opera2 IN puccini.ltm
WHERE
    ...
```
Add support for non-existential queries

This still needs to be fleshed out...

SELECT $composer WHERE
WHERE
  composed-by( $opera : opera, $composer : composer ),
  EVERY $opera/premiere-date > "1900"
?
Constructors for TMs (1)

(Short problem description)

WHERE

$city : city,
{ premiere($opera, $city) |
  premiere($opera, $theatre),
  located-in($theatre, $city) }
ORDER BY $opera DESC?

Would normally return cities, operas, and theatres
…but no associations (or names or occurrences)

What should the TM contain in addition to those topics?
  – premiere and located-in associations?
  – names? types? locators?
Why not just specify exactly that…
Constructors for TMs (2)

(Simple solution example)

FOR $opera, $city, $theatre
WHERE
$city : city,
{ premiere($opera, $city) |
premiere($opera, $theatre),
located-in($theatre, $city) }
ORDER BY $opera DESC?
RETURN
$*/@bn,
premiere(),
$opera/premiere-date,
located-in()

plus:
• all basenames for these topics
• all premiere assocs of these topics
• all premiere-date occs (of these)
• all located-in assocs (of these)
Constructors for TMs (3)

(variant for all associations and all basenames/occurrences)

FOR $opera, $city, $theatre  
return all these topics
WHERE
  $city : city,
  { premiere($opera, $city) | 
    premiere($opera, $theatre),
    located-in($theatre, $city) } 
ORDER BY $opera DESC?
RETURN
  *(),
  $opera/*,
  $city/*,
  $theatre/*

plus:
  ▪ all assocs involving these topics
  ▪ all occs & bns of these topics
  ▪ all occs & bns of these topics
  ▪ all occs & bns of these topics
Constructors for TMs (4)

Transformation to another ontology:

FOR $opera, $city
WHERE
   $city : city,
   { premiere($opera, $city) | premiere($opera, $theatre), located-in($theatre, $city) }
ORDER BY $opera DESC?
RETURN
myOnt:first-performed($opera : myOnt:work, $city : myOnt:place)
Other stuff

In addition, retain the following from tolog:

- prefixes (namespaces for identifiers and locators)
- modules (user-defined predicates, functions, etc.)
- inference rules
Prefixes

Need a way to use namespaces for URIs, similar (but not necessarily identical) to that already in tolog:

```sql
USING opera FOR i"http://psi.ontopia.net/opera/
USING geolang FOR i"http://psi.oasis-open.org/geolang/
SELECT $city, count($opera)
WHERE
  $city : opera:city,
  { opera:premiere($opera, $city) |
    opera:premiere($opera, $theatre),
    geolang:located-in($theatre, $city) }
ORDER BY $opera DESC?
```
Unaddressed “weaknesses”

The following issues, identified as weaknesses in one or more of the prototype implementations, have not yet been addressed:

- Typing for primitive values (TMCL?)
- Scope (not yet fully cooked in this model)
- Set support (needs clarifying)
- Namespace declarations/identity syntax (not finalized)
- Integration of functions (not fully worked out)
- Transformation of TMs (not fully worked out)
- Non-existential queries (not fully worked out)
- Relationship to TMCL (not yet completely clear)
Schedule for further action

larsbot -> china (April 22 - May 18)
\rho does first draft in peace
WG3/TMQL-RG meets late summer (Montreal? Europe? Oslo?)

Target dates:
First WD submitted to SC34 by 2004/10/01
  ▪ In time for NB review prior to next SC34 meeting
Approved for CD ballot 2004/11
  ▪ Anticipated SC34 meeting in Nov/Dec 2004 timeframe
Second WD submitted by 2005/03/18
  ▪ In time for NB review prior to Spring 2005 meeting
Approved for FCD ballot 2005/05
  ▪ (Unless first ballot leads to extensive changes)