TMQL issues

October 14, 2007

Support for transitive closures

- **We think TMQL needs to support transitive closures on query expressions**
- **Something like**
  - root-node (<- parent [^parent-child] -> child)*
- **Issues**
  - do we want to be able to require at least one step?
  - do we want to be able to control whether the start node is included?
  - do we want to be able to say repeat 1..3 times (instead of arbitrary)?
- **LMG will write up a proposal**
  - and requirements
  - deadline is November 12

Clause 3.1: Syntax Conventions

- **Could we move all this out to CTM, and just do what CTM does?**
  - no
  - *but* must align with CTM so that there are no inconsistencies

Clause 3.3: Ontological commitments

- **Couldn't we move the first part of this to the TMDM-TMRM mapping?**
  - no, but we reference the mapping and make this into a note

Clause 4.1: Constants

- **[1] constant ::= atom | item-reference**
- **[2] atom ::= iri | ...**
- **[9] iri ::= < QIRI > | QIRI**
- **[16] item-reference ::= identifier | QIRI**
- **Couldn't an IRI reference be either one of these?**
  - how does a parser know how to interpret an IRI reference?
  - seems like if QIRI appears via atom the result is an IRI,
  - whereas if it appears via item-reference, it's an information item
- **However, production [9] has to change anyway**
  - so Robert will change this, and make sure this isn't an issue

Clause 4.1: Constants

- **[9] iri ::= < QIRI > | QIRI**
  - do we really need two variants of this syntax?
  - no, we don't, and this is already removed in CTM, so we follow suit here

Clause 4.3: Item References

- **[16] item-reference ::= identifier | QIRI**
- **[11] QIRI ::= IRI | QName**
- **The text in 4.3 only defines interpretation of**
  - identifiers, and
  - QNames
- **Nothing is said about IRIs**
  - actually, it's said in the part about QNames, but it's easy to miss
  - however, could add a sentence to make this clearer

Clause 4.2: Atoms

- **What is “undef” for?**
- **The use case is to express a query like**
  - give me all people and their home page URIs, if they have one
  - select $p / name, $p / homepage || undef
    from ... where $p isa person
- **CTM calls it “null”**
  - need to consider whether it should be in CTM
  - also whether to align TMQL
  - note that XML Schema uses “nil”...
- **TMQL also has a “null”**
  - this means the same as (), that is the empty sequence
  - this is actually what Lisp calls “nil”
  - we change this to “nil”
  - however, “undef” stays put
  - CTM either removes “null” or changes the name to “undef”

Clause 4.2: Atoms

- **Do we need “true” and “false”?**
- **They are there because CTM has them...**
  - they are slightly confusing
  - select $p / name where $p isa person & false
  - this query actually works, because “false” produces a value (even if that value is false)
- **There is general agreement that we’d better off without them**
  - so if CTM removes these, TMQL will do the same
  - names must be the same as in CTM

Clause 4.2: Atoms

- **Don’t we need to define the productions for date and dateTime?**
  - waiting for CTM to define these datatypes, will then reference CTM
  - actually, planning to reference *all* datatype definitions from CTM instead of reproducing here

Clause 4.2: Atoms

- **We must support**
  - Unicode character references in strings (a la \u00DE)
  - some escape sequences like \n and \r
- **We all agree on this**
  - CTM has this, but only 4 digits, must extend to also support 8 (unambiguously)
  - TMQL will just inherit this string syntax

Clause 4.3: Item References

- **Clause 4.3 says:**
  - "If the item reference is an identifier then this identifier is interpreted as an item identifier (TMDM, clause 5.1) for an item.
  - EXAMPLE:
    - jack / name"
- **But item identifiers are URIs, and these are simple names**
  - so how does this work?
  - it doesn't actually work, since the [base locator] property was removed from TMDM
  - we don't actually want to encourage this,
    - so we prefer to make bare identifiers be interpreted using a default prefix
  - we still need to be able to reference via item identifiers
    - this *is* possible via the item-identifier axis
    - however, we also need a shorthand
    - we appropriate the longhand ~ for subject identifiers and now use it for item identifiers (unless Robert comes up with something better)

Clause 4.7: Composite content

- **[K] content ::= path expression**
- **==> { path expression }**
- **If the curlies are not necessary, why have this?**
  - the curlies are necessary in many cases to avoid syntactical amibiguity
  - the canonical form of the syntax is with curlies (for some reason)
  - therefore the shorthand goes this way

Clause 4.10: Topic Maps content

- **The syntax of embedded CTM is not specified**
  - need to do this formally by modifying some of the CTM productions
  - CTM has to stabilize first (in the meantime we put in a placeholder)
- **Should it be possible to globally define CTM templates somewhere?**
  - so they don't have to be repeated in each TM-constructing query
  - no
- **We also get into trouble with bare identifiers here...**
  - there is no base URI against which these are interpreted
  - they therefore cause errors if there is no default prefix
  - if there is a default prefix they became subject identifiers (as in CTM)

Clause 5.4: Variable Assignments

- **“The function fn:concat takes a tuple sequence and produces one which consists only of the concatenation of all the original tuples.”**
  - @p in fn:concat(// person)
- **How does this actually work?**
  - and where is it specified?
- **We will revisit this and make it more explicit how it works...**

Clause 6.3: Environment Clause

- **There is no defined syntax for prefixes here**
  - that means TMQL cannot actually be used
- **We *have to* define a syntax for defining prefixes**
  - we use the CTM syntax for this
- **We also have to decide what to do with the clause itself**
  - we decided to take out the syntax for the environment clause
    - that is, we change the right-hand side of production 46 to be declaration of prefixes

Clause 6.4: SELECT Expressions

- **The shape of these is**
  - select ...
  - from ...
  - where ...
  - order by ...
  - unique
  - offset ...
  - limit ...
- **Why is the “unique” not just after “select”?**
  - because it operates at that point in the query execution
  - so we leave it where it is

The syntax is too complex

- **What can we do to fix that?**
  - root-node <- parent [^parent-child] -> child
- **Procedure**
  - identify the main cases where it can be simplified
  - write up proposals for simplification
  - discuss these offline on the mailing list before Kyoto
  - any cases not settled by Kyoto get voted on there
- **Exhortations**
  - everyone: *please* study the syntax!
  - developers: download perlxtm from SourceForge (or TM from CPAN)

What should be moved to the mapping?

- **Robert said parts of this spec should be moved into the TMRM-TMDM mapping**
  - so maybe we should do that
  - but which parts?
  - this was a misunderstanding

Next actions

- **LMG to write new draft with changes given here before November 12**
- **Plus other proposals etc**