

# Compact Syntax for Topic Maps (CTM)

-

## initial work

Professor Sam G. Oh,  
Sung Kyun Kwan University;  
Gabriel Hopmans, Morpheus software;

# This presentation

- Contains sheets that are not necessarily to be discussed in detail
  - Some are more to get a common understanding
  - The most important ones are about the requirements

# Agenda

1. Introduction why we need CTM
2. Discussion on requirements
  - goals, requirements and issues
3. Evaluations on LTM and AsTMA=
  - This is something for next actions
4. Discussion on tasks/proceeding with the CTM work item

# To start with..

- Main point: CTM is for manual creation; real end-users will use editing interfaces
- CTM is more for technies to make their life easier
- CTM is relegated to a kind of "supporting role", as "nice syntax for crazy techies"
- XTM stays the basis for interchange
  - CTM will be an add-on

# Background

- In review of TMQL proposals it was recognised that Robert Barta had designed a language family with consistent syntax, instead of creating three separate languages (compact topic map syntax, schema syntax, query syntax).
  - Is a major benefit
  - Example scope, indicated in the same way in all three languages.
- ISO is only doing two of these languages: a schema language and a query language.
  - But the query language is going to have an update part
    - .. continued

# Better 3 syntaxes in parallel

- Typically, update languages have three operations:
  - INSERT: add new data
  - DELETE: remove data
  - UPDATE: change data
  - To support the INSERT operation, we will be forced to provide some way to express the topic map information to be added.
- Ability to express the characteristics of particular topic, so that you can write something like:
  - INSERT <topic-map-fragment-goes-here>
    - textual syntax for this is preferred
- Thus better have design for TMQL for querying, TMCL for schemas, and textual syntax in parallel

# Use of CTM

- For TM inference language
- For a TM construction language in TML part 1
  - Maybe a superset of CTM
- Human readable examples
  - For papers and presentations

# Thus why CTM?

- Shorthand syntax for developers
  1. providing a common, lightweight syntax as a basis for TMCL and TMQL
  2. providing human-readable examples
    - Illustration in papers, examples
  3. manually authoring topic maps
    - But not to replace XTM, just for fragments and within TMCL/TMQL



# Why CTM?

- We need the syntaxes to be coherent,
  - and so CTM needs to happen now, rather than after finalising TMQL-1/TMCL
  - Avoid the use delimiters for the wrong things.
  - do CTM as a prelude to nailing down TMQL syntax

# Goals

- Goal A: CTM should not stand in the way of TMCL/TMQL syntactically.
- Goal B: CTM needs to be easy to write.
- Goal C: CTM needs to be easy to read.
- Goal D: CTM should integrate well with the graphical notation for topic maps, if there ever is one.
  - So this will on long term
- Goal E: CTM needs to be easy to learn and teach.
- Goal F: goal: CTM needs to be easy to see the big picture. When editing a topic map CTM can make it possible to show as many as topics and/or associations in the same window at a time. It makes my work efficient.

# 2. Requirements

- Divided into five clusters:
  - 2.1 On general use (3)
  - 2.2 CTM features and things it needs to support (4)
  - 2.3 Use in other parts (3)
  - 2.4 ISO specific (4)
  - 2.5 CTM and TMDM (2)
    - Some have issues
- Some initial requirements are changed into goals

# 2.1 General Use

- **2.1.1 CTM should be optimized for humans**
  - more frequently used features should be MUCH shorter writeable than non-frequent ones. On the other hand this should be balanced with the use of 'strange' characters.
- **2.1.2 It should be possible to add author comments within a CTM file, and support the "commenting-out" of CTM constructs. (support for goal C & F)**
- **2.1.3 make it possible to reformat CTM on slides and in printed documents, so that it can fit the allotted space**

## 2.2 features and support (1)

- **2.2.1 CTM must support directives of existing proposals like:**
  - PREFIX,
    - To use references/namespaces for topic id's and locators
  - MERGEMAP and INCLUDE
    - To merge topic maps into the file
  - Version
    - Which specification version the instances uses

## 2.2 features and support (2)

- **2.2.2 CTM shouldn't be too hard to parse.**
- A syntax that makes it easy to cut-and-paste and move chunks around without breaking things
- **2.2.3 CTM has to support all character encodings.**
- **2.2.4 CTM needs to have an escape syntax for Unicode characters**

## 2.3 with other parts

- **2.3.1 Must be syntactically/conceptually aligned with the TMCL and TMQL syntaxes.**
- At the moment we only have a TMQL syntax, and that can change, but, for example, if TMQL uses @ for scope, CTM probably should, too, or they should both change.
- **2.3.2/2.3.3 CTM for operations in TMQL, part 2**
  - (2.3.2) It must be possible to use CTM for the "INSERT" operation in TMQL, part 2. This will give the ability to express the topic map information to be added.
  - (2.3.3) We may also want to use CTM for the "UPDATE" operation in TMQL, part 2, but this is not necessarily a given.
  - Issue: continued

# CTM TMQL 2 issue

- Kal: | 1) If TMQL does not support update using XTM, will that alienate those users who prefer XTM to a compact syntax? How much effort would it be for TMQL to support both syntaxes for update operations?

Lars: Those are good questions, and I have to confess I don't fully know the

answers to them. Whether a non-XTM syntax for additions will alienate users is difficult to answer because these users will already be working inside a non-XML syntax (TMQL). Most likely something like

```
INSERT <topic id="foo">...</topic>
```

is one thing, but TMQL also has topic map constructors for producing topic map output, and quite possibly also TM->TM transforms, which means CTM gets mixed quite deeply into TMQL. So I don't really know; those users who prefer writing their topic maps in XTM might be able to help us here.

As for the effort in making TMQL support both: I think for the simple INSERT case it will be pretty straightforward. For the more complex case of topic map output construction I'm not sure. We might be able to get away with using the XML construction and then having a mode to interpret XTM output as topic map information.



## 2.4 ISO / spec - specific

- **2.4.1 CTM should encourage use of PSIs to refer to topics over use of IDs.**
- **2.4.2 CTM must support IDs for topic map constructs**
- **2.4.3 The CTM project should \*not\* create a CTM test suite.**
  - (This is not ISO work, and should be left to others.)
- **2.4.4 The CTM specification must use EBNF to unambiguously define the syntax of CTM.**

# CTM and TMDM

- **2.5.1 The CTM specification must define deserialization of CTM to TMDM.**
- **2.5.2 CTM MUST fully represent TMDM.**
  - note: one previous requirement: "CTM must support embedded markup" is implied by this)
  - **Issue: continued, next slide**

## 2.5.2 CTM must fully represent TMDM

- CTM will be a notation for Topic Maps and not something like a "linear XTM notation".
  - CTM is not so much XTM related but more TMDM related.
- We want a notation for topics without an ID
  - just using an subject locator / subject identifier or some notation to tell the parser "I do not care about the ID"

# Considerations

- A 'macro' feature to make it possible to write associations more compactly
- Indicate 'sections' within a CTM file
  - sections would be things like "all topic types", "all role types"

# 3. Evaluation of proposals

- Evaluate existing proposals (LTM and AsTMA=) against requirements and..
- Some general evaluation criteria:
  - conciseness of the language
  - syntax to organise TM content
  - technical issues or background one needs to have
  - compactness
  - the reduction of typing efforts
  - being part of a coherent language family

# CTM, LTM and AsTMa=

- XTM, LTM and AsTMa= comparisons
  - Tables with characteristics, advantages, disadvantages, similarities etc..
- CTM vs XTM features
- PTM, a Python-inspired syntax

## 4. CTM ISO 13250 part 6?

- How to proceed?
  - evaluations
  - Use cases and results in all the syntaxes
- Editors?