

Creating a Topic Map

Not as scary as you thought...

Steve Pepper

pepper.steve@gmail.com

Topic Maps 2008, 2008-04-02

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

Today's agenda

- **History of Topic Maps**
 - Davenport, ISO, TopicMaps.Org, OASIS
- **Syntaxes**
 - XTM, LTM and CTM
- **Demo: Creating a topic map**
- **Ontology driven editing**

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

The History of Topic Maps

Davenport
ISO
TopicMaps.Org
OASIS

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

The original insight (1991)

- **Davenport Group**
 - Early days, before development of DocBook
 - IT consortium of people interested in hardware and software documentation
 - Business requirement to merge (electronic) back-of-book indexes
 - Digital Equipment Corp. (DEC) wished to bundle O'Reilly's UNIX documentation with their UNIX systems
 - They wished to create a master index of all hardware and software documentation by merging the individual indexes
 - The attempt was a miserable failure – but why?
- **The key insight**
 - Back-of-book indexes are knowledge structures
 - “Indexes, if they have any self-consistency at all, conform to models of the structure of the knowledge available in the materials that they index. But the models are implicit, and they are nowhere to be found! If such models could be captured formally, then they could guide and greatly facilitate the process of merging modelled indexes together.” (Steve Newcomb)

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

HyTime and CApH

- **Merging of indexes was the original challenge**
 - In order to solve it, a model was devised based on the new HyTime standard
 - HyTime = ISO 10744 *Hypermedia and Time-based Structuring Language*, a forerunner of Xlink, XPath, etc.
 - The model was called Topic Navigation Maps
 - Main activity of CApH (Conventions for the Application of HyTime), whose purpose was to develop concrete use cases for HyTime
 - The model went through several iterations before being introduced to ISO

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

ISO 2000 – 1st Edition

- **Accepted as New Work Item in ISO in 1996**
 - Editors: Michel Biezunski and Martin Bryan
 - More iterations, further development of the model
 - Editors joined by Steve Newcomb
- **Finalized in 1999**
 - Approved and published as ISO/IEC 13250:2000
 - Defined in terms of an SGML DTD
 - (Actually a meta-DTD, as defined in the Annex on SGML Architectures in HyTime)
 - Linking model based on HyTime (varlink, anchspec)
 - Included a construct called facets
 - Purpose: “to apply metadata/value pairs to information resources”

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

TopicMaps.Org 2001

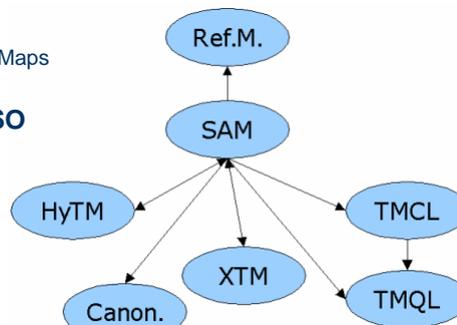
- **User consortium formed in January 2000**
 - Initiativ taken by Steve Newcomb and Michel Biezunski
- **Goal: To adapt Topic Maps for the Web**
 - In particular: Develop an alternative syntax based on web standards, i.e. XML and Xlink
- **XML Topic Maps (XTM) 1.0 specification**
 - Approved and delivered in March 2001
 - Editors: Steve Pepper, Graham Moore
 - XML-based syntax and Xlink-based linking model
 - Introduced distinction between direct and indirect identification
 - Removed concept of facets as no longer necessary
 - Anything can be a subject, including information resources

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

ISO 2003 – 2nd Edition

- **XTM added to ISO 13250 in 2003**
 - ISO/IEC 13250:2003 – Topic Maps (2nd Edition)
- **Further development in ISO**
 - New version of ISO 13250 to include
 - Data model (SAM)
 - Reference model
 - Canonicalization
- **Two new standards**
 - Constraint Language (TMCL)
 - Query Language (TMQL)



From ISO/IEC JTC 1/SC34 N323
2002-06-23

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

ISO 2007 – Multipart standard

- **ISO/IEC 13250: Topic Maps**

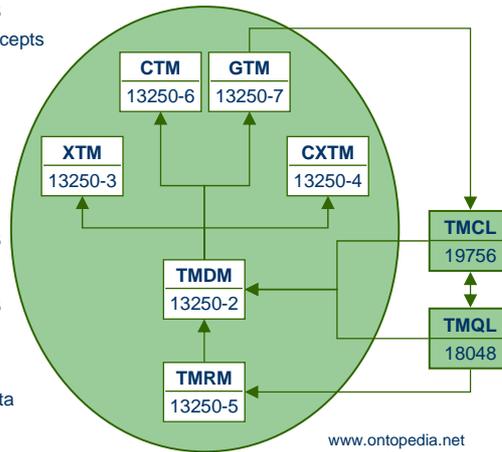
- Part 1: Overview and Basic Concepts
- Part 2: Data Model
- Part 3: XML Syntax
- Part 4: Canonicalization
- Part 5: Reference Model
- Part 6: Compact Syntax
- Part 7: Graphical Notation

- **ISO/IEC 18048: Topic Maps Query Language**

- **ISO/IEC 19756: Topic Maps Constraint Language**

- **ISO/IEC TR 29111:**

- Expressing Dublin Core Metadata Using Topic Maps



<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

Further plans

- **Dublin Core**
- **RDFTM Interoperability**
- **Published Subjects**
- **Remote Access Protocol**
- **RDBMS mapping**
- ...
- ...

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

Syntaxes

XTM, LTM and CTM
What are they?
When should I use which?

Topic Maps Syntaxes

- **HyTM** (HyTime Topic Maps)
 - Original syntax, expressed in terms of SGML and HyTime
 - *No longer part of ISO 13250*
- **XTM** (XML Topic Maps Syntax)
 - Later, XML-based syntax, recently moved to version 2.0
 - *Easy to understand but very verbose*
- **LTM** (Linear Topic Map Notation)
 - Defined by Ontopia in 2001 and supported by other products
 - *A simple ASCII syntax for rapid prototyping*
- **CTM** (Compact Topic Maps Syntax)
 - ISO standard replacement for LTM
 - *Complete draft exists, but no implementations yet*

Topic Map – XTM 1.0 Syntax

```

<!ELEMENT topicMap
  ( topic | association | mergeMap )* >
<!ATTLIST topicMap
  id          ID          #IMPLIED
  xmlns      CDATA      #FIXED 'http://www.topicmaps.org/xtm/1.0/'
  xmlns:xlink CDATA      #FIXED 'http://www.w3.org/1999/xlink'
  xml:base   CDATA      #IMPLIED >

<?xml version="1.0" encoding="ISO-8859-1"?>
<topicMap
  xmlns="http://www.topicmaps.org/xtm/1.0/"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  >

  <!-- topics, associations, and mergeMap elements go here -->

</topicMap>

```

Topic Map – LTM Syntax

```

/* topics, associations, and occurrences go here */

```

Topic – XTM 1.0 Syntax

```
<!ELEMENT topic
  ( instanceOf*, subjectIdentity?, ( baseName | occurrence )* )
>
<!ATTLIST topic
  id ID #REQUIRED
>
```

```
<topic id="italy">
```

```
...
```

```
</topic>
```

```
<topic id="puccini">
```

```
...
```

```
</topic>
```

Topic – LTM Syntax

```
[topic-id]
```

```
[italy]
```

```
[puccini]
```

Topic Name – XTM 1.0 Syntax (1 of 2)

```

<!ELEMENT baseName ( scope?, baseNameString, variant* ) >
<!ATTLIST baseName
  id          ID          #IMPLIED >

<!ELEMENT baseNameString ( #PCDATA ) >
<!ATTLIST baseNameString
  id          ID          #IMPLIED >

<!ELEMENT variant ( parameters, variantName?, variant* ) >
<!ATTLIST variant
  id          ID          #IMPLIED >

<!ELEMENT variantName ( resourceRef | resourceData ) >
<!ATTLIST variantName
  id          ID          #IMPLIED
>

```

Topic Name – XTM 1.0 Syntax (2 of 2)

```

<topic id="la-scala">
  <baseName>
    <baseNameString>Teatro alla Scala</baseNameString>
    <variant>
      <parameters>
        <subjectIndicatorRef
          xlink:href="http://www.topicmaps.org/xtm/1.0/core.xtm#display"/>
      </parameters>
      <variantName>
        <resourceData>La Scala</resourceData>
      </variantName>
    </variant>
    <variant>
      <parameters>
        <subjectIndicatorRef
          xlink:href="http://www.topicmaps.org/xtm/1.0/core.xtm#sort"/>
      </parameters>
      <variantName>
        <resourceData>scala, teatro alla</resourceData>
      </variantName>
    </variant>
  </baseName>
</topic>

```

Topic Name – LTM Syntax

```
[topic-id = basename; sortname?; dispname?]
```

```
[la-scala = "Teatro alla Scala"; "scala, teatro alla"; "La Scala"]
```

Topic Type – XTM 1.0 Syntax

```
<instanceOf> subelement
```

```
<topic id="opera">
  ...
</topic>
```

```
<topic id="tosca">
  <instanceOf>
    <topicRef xlink:href="#opera"/>
  </instanceOf>
</topic>
```

```
<topic id="boito">
  <instanceOf>
    <topicRef xlink:href="#composer"/>
  </instanceOf>
  <instanceOf>
    <topicRef xlink:href="#librettist"/>
  </instanceOf>
</topic>
```

Topic Type – LTM Syntax

```
[topic-id : topic-type]

[tosca : opera]

[boito : composer librettist]
```

Occurrence – XTM 1.0 Syntax

```
<occurrence> subelement:
external/internal resources: <resourceRef> or <resourceData>

<!ELEMENT occurrence
  ( instanceOf?, scope?, ( resourceRef | resourceData ) )
>
<!ATTLIST occurrence
  id ID #IMPLIED
>

<topic id="la-boheme">
  <occurrence>
    <instanceOf><topicRef xlink:href="#homepage"/></instanceOf>
    <resourceRef
      xlink:href="http://www.opera.it/Opere/La-Boheme/La-Boheme.html"/>
    </occurrence>
    <occurrence>
      <instanceOf><topicRef xlink:href="#premiere-date"/></instanceOf>
      <resourceData>1896 (1 Feb)</resourceData>
    </occurrence>
  </topic>
```

Occurrence – LTM Syntax

```
{topic-id, occurrence-type, [URL | data]}
```

```
{la-boheme,
  homepage, "http://www.opera.it/Opere/La-Boheme/La-Boheme.html"}
{la-boheme, premiere-date, [[1896 (1 Feb)]]}
```

Topic – “Complete” XTM 1.0 Syntax

```
<topic id="la-boheme">
  <instanceOf><topicRef xlink:href="#opera"/></instanceOf>
  <baseName>
    <baseNameString>La Bohème</baseNameString>
    <variant>
      <parameters>
        <subjectIndicatorRef
          xlink:href="http://www.topicmaps.org/xtm/1.0/core.xtm#sort"/>
      </parameters>
      <variantName><resourceData>Boheme</resourceData></variantName>
    </variant>
    ...
  </baseName>
  <occurrence>
    <instanceOf><topicRef xlink:href="#homepage"/></instanceOf>
    <resourceRef
      xlink:href="http://www.opera.it/Opere/La-Boheme/La-Boheme.html"/>
    </occurrence>
    ...
</topic>
```

Topic – Complete LTM Syntax

```
[la-boheme : opera
  = "La Bohème"; "Boheme"
  = "The Bohemian Girl"; "Bohemian Girl, The" /english]

{la-boheme,
  homepage, "http://www.opera.it/Opere/La-Boheme/La-Boheme.html"}
{la-boheme,
  premiere-date, [[1896 (1 Feb)]]}
```

Association – XTM 1.0 Syntax

```
<!ELEMENT association (instanceOf?, scope? , member+)>
<!ATTLIST association
  id ID #REQUIRED>
<!ELEMENT member (roleSpec?, (topicRef | ...)+) >
<!ATTLIST member
  id ID #IMPLIED>
<!ELEMENT roleSpec (topicRef | ...) >

<association>
  <instanceOf><topicRef xlink:href="#composed-by"/></instanceOf>
  <member>
    <roleSpec><topicRef xlink:href="#composer"/></roleSpec>
    <topicRef xlink:href="#puccini"/>
  </member>
  <member>
    <roleSpec><topicRef xlink:href="#work"/></roleSpec>
    <topicRef xlink:href="#tosca"/>
  </member>
</association>
```

Association – LTM Syntax

```
assoc-type ( [role-player] : role-type, [role-player] : role-type )
```

```
composed-by( puccini : composer, toska : work )
born-in    ( puccini : person,  lucca : place )
```

```
composed-by( puccini, toska )
born-in    ( puccini, lucca )
```

Note: When omitted, the role type will be assumed to be identical to the type of the role playing topic. This can be a useful short-hand, but it is not always what you want.

Subject Identity – XTM 1.0 Syntax

```
<!ELEMENT topic (instanceOf*, subjectIdentity?,...)>
<!ELEMENT subjectIdentity (resourceRef?, (topicRef | subjectIndicatorRef)*) >

<!-- Refer to a resource as subject: -->
<topic id="foo">
  <subjectIdentity>
    <resourceRef xlink:href="http://www.ontopia.net"/>
  </subjectIdentity>
  <baseName>
    <baseNameString>The Ontopia Website</baseNameString>
  </baseName>
</topic>

<!-- Refer to a subject indicator: -->
<topic id="bar">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href="http://www.ontopia.net/about.html"/>
  </subjectIdentity>
  <baseName>
    <baseNameString>Ontopia</baseNameString>
  </baseName>
</topic>
```

Subject Identity – LTM Syntax

```
[topic-id = names %subject-address-URL]
[topic-id = names @subject-indicator-URL]

/* Refer to a resource as subject: */
[foo = "The Ontopia Website" %"http://www.ontopia.net" ]

/* Refer to a subject indicator: */
[bar = "Ontopia" @"http://www.ontopia.net/about.html"]
```

Scope – XTM 1.0 Syntax

```
<!-- "scope" subelements on baseName, occurrence, and association
      (also "parameters" on variantName) -->

<topic id="composed-by">
  <baseName>
    <baseNameString>composed by</baseNameString>
  </baseName>
  <baseName>
    <scope><topicRef xlink:href="#composer"/></scope>
    <baseNameString>composer of</baseNameString>
  </baseName>
</topic>

<topic id="la-boheme2">
  <baseName>
    <baseNameString>La Bohème (Leoncavallo)</baseNameString>
  </baseName>
  <baseName>
    <scope><topicRef xlink:href="#leoncavallo"/></scope>
    <baseNameString>La Bohème</baseNameString>
  </baseName>
</topic>
```

Scope – LTM syntax

(name or occurrence or association) / **scoping-topic(s)**

```
[born-in = "composed by"  
  = "composer of" / composer ]  
  
[la-boheme1 = "La Bohème (Puccini)"  
  = "La Bohème" / puccini ]  
  
[la-boheme2 = "La Bohème (Leoncavallo)"  
  = "La Bohème" / leoncavallo ]
```

Demo: Creating a topic map

Ontology-driven editing

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

What is ontology? (philosophy)

"In philosophy, ontology (from the Greek *ον* = being and *λόγος* = word/speech) is the most fundamental branch of metaphysics. It studies being or existence as well as the basic categories thereof – trying to find out what entities and what types of entities exist. Ontology has strong implications for the conceptions of reality."

<http://en.wikipedia.org/wiki/Ontology>

- **Originally a branch of metaphysics (or philosophy)**
 - The study of being, or existence
 - Founded by Plato and Aristotle
- **Deals with questions such as**
 - "What exists?" (for example, does the idea of a "horse" exist?)
 - "What are subjects, objects, and their relationships?"
- **Has a specialized meaning in informatics**
 - Among other things, it is possible to talk of an ontology

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

What is an ontology? (informatics)

"In both computer science and information science, an ontology is a data model that represents a set of concepts within a domain and the relationships between those concepts. It is used to reason about the objects within that domain. Ontologies are used in artificial intelligence, the semantic web, software engineering, biomedical informatics and information architecture as a form of knowledge representation about the world or some part of it."

[http://en.wikipedia.org/wiki/Ontology_\(computer_science\)](http://en.wikipedia.org/wiki/Ontology_(computer_science))

- **Used (and abused) to mean almost any form of conceptual classification scheme**
 - Ranges from simple type hierarchies to more complex models; should also include other semantic relations describing relationships between concepts
- **In Topic Maps an ontology is**
 - "The set of typing topics that is used within a topic map, or that defines a class of topic maps"
 - May or may not include the constraints that apply to those typing topics

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

What is ontology-driven editing?

- **A user-friendly way to create topic maps!**
- **The principle is simple**
 - The ontology describes what kind of things can exist in the topic map
 - It also includes constraints on
 - Which types of statement are used with which types of topics
 - What cardinality they have
 - Based on this, the interface is automatically configured for data entry
- **The benefits**
 - Easier user interface – no need to understand syntax
 - More consistent topic maps
- **Ontopoly is such an editor**

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

How to use Ontopoly

1. **Read the Ontopoly User Guide!**
 - It will save you a lot of grief in the long run
2. **Start the program from OKS Samplers**
 - Open an existing Ontopoly topic map
 - Import an existing non-Ontopoly topic map
 - Or create a new topic map
3. **Use the Description tab to describe the topic map**
 - Also to validate it and a few other things
4. **Use the Ontology tab to define the ontology**
 - topic types, type hierarchy, association types, role types, name types, occurrence types
 - "fields" (names, identifiers, occurrences, and associations) that apply to each topic type, their order and cardinality
5. **Use the Instances tab to populate the data**
 - Uses an automatically configured forms-based interface

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

Some tips on ontology creation

- **When starting a new Ontopoly topic map**
 - Sketch out the basic ontology on paper before using Ontopoly
 - Create the type hierarchy first. Keep it simple.
 - Create association types and role types (where necessary)
 - Specify what the role-playing topic types are
 - Create occurrence types and name types
 - Go to each topic type in turn, starting at the top of each type hierarchy, and assign additional fields
 - Make sure you are satisfied with the ontology before you add too much data, otherwise it can be a lot of work to change things
- **When importing an existing non-Ontopoly topic map**
 - Check name types and occurrence types first
 - Then check association types (and role types)
 - Then check each topic type in turn (top down)

<http://www.topicmaps.com/tm2008/pepper3.ppt>

www.ontopedia.net
psi.ontopedia.net

Some comments on Ontopoly

- **Takes a little getting used to**
 - But it's well worth the effort
- **Does not (yet) support scope or variant names**
 - Use typed names instead of scoped names
- **Does not allow topics to have multiple types**
 - Except for supertypes
- **Includes system information in the topic map**
 - The topic map can be exported without this information
 - It can be hidden in the Omnigator
 - Customize → Nontopoly model
- **Important points to remember**
 - Clicking on any link submits the HTML form, but **does not save to disk**
 - You **MUST** click on the **Save** button regularly