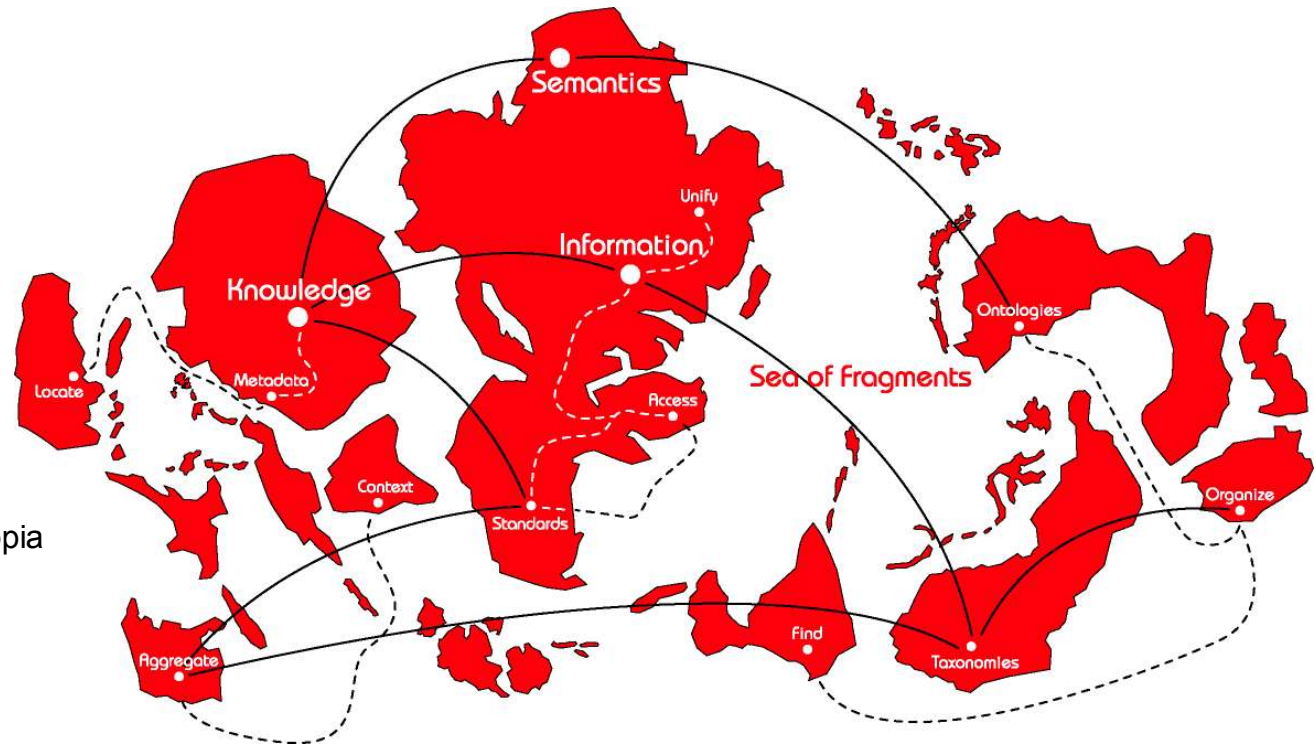


# Overview of TMDM and XTM 1.1

---

## Standards review, part 1



**Lars Marius Garshol**

Development Manager, Ontopia  
 <larsga@ontopia.net>

2005-03-29

# Agenda

---

- **Brief historical background**
- **TMDM walkthrough**
- **XTM 1.1 walkthrough**
- **Discussion**

*Slides titled "Comments" contain comments from the participants*

## Goals for this evening

---

- To inform interested users about the standards
- To get feedback before the Amsterdam meeting in May

---

## Historical background



*ISO SC34 WG3*

*XTM 1.0*

*TMDM and XTM 1.1*

# Topic Maps and ISO

---

- **ISO accepted Topic Maps as a work item in 1996 and assigned it to SC34 (the SGML/DSSSL/HyTime committee)**
  - Michel was the original editor, joined by Martin Bryan and Steve Newcomb
  - Finally submitted for ballot in October '98, and published in January 2000 as ISO/IEC 13250
- **ISO/IEC 13250:2000 Topic Maps**
  - Describes the basic model and constructs
  - Defines a HyTime- and SGML-based syntax known as HyTM
  - The model was good (except that it was defined in terms of syntax)
- **The syntax had three main problems:**
  - It was SGML, no XML support at all
  - It used HyTime for all linking, so no URIs
  - It used architectural forms, so no fixed syntax

## TopicMaps.Org

---

- **To create a syntax for use on the Web, TopicMaps.Org was set up outside ISO**
  - A number of people worked intensively for a year on creating XTM
  - XTM solves all three problems (see previous slide) with HyTM
- **XTM also improved the model:**
  - The distinction between addressable and non-addressable subjects was introduced
  - Topics could now represent resources (so facets were no longer required)
  - Sort names and display names were generalized (as variant names)

## Further work on ISO 13250

---

- **ISO/IEC 13250:2003 now includes XTM 1.0**
- **Two new projects have been approved:**
  - TMQL: a query language for topic maps (ISO 18048)
    - Committee Draft published February 2005
  - TMCL: a schema language for topic maps (ISO 19756)
    - Committee Draft published February 2005
- **A formal data model became necessary to answer questions like:**
  - How can TMQL query both syntaxes?
  - How are the two syntaxes related?
  - How should the syntaxes be interpreted?

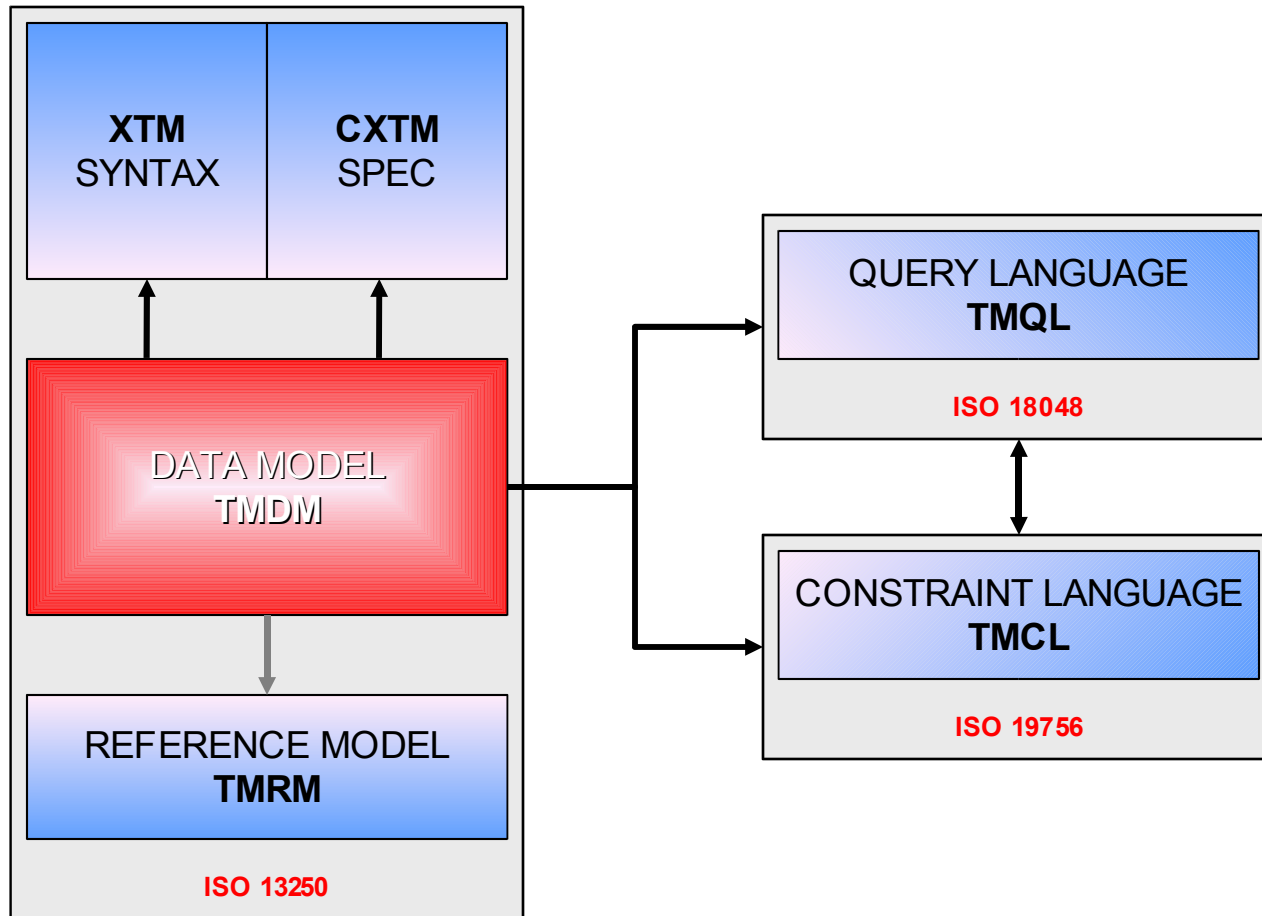
# The new ISO 13250

---

- **Will be a multi-part standard, consisting of**
  - Part 1: Overview of Basic Concepts
  - Part 2: Data Model
  - Part 3: XML Syntax (ie: XTM)
  - Part 4: Canonical XTM
- **May also contain**
  - Part 5: Reference Model (conceptual, high-level + mathematical model)
- **Will no longer contain**
  - A HyTM syntax Specification
- **Timetable**
  - Final Committee Drafts of parts 2-4 published January 2005
  - Part 5 being worked on; current draft lacks math part

# Roadmap to the TM standards

---



---

## TMDM



*Overview*

*The information items*

*The merging rules*

*The new PSIs*

*Notable issues*

# The Topic Maps Data Model (TMDM)

---

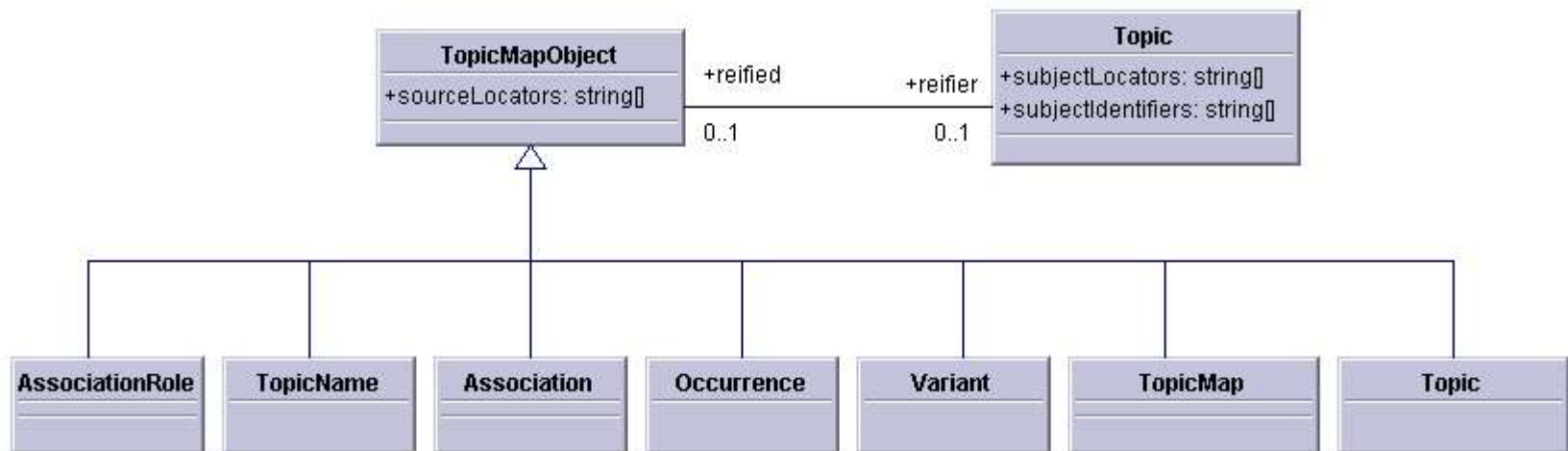
- **Created to define meaning and structure of topic maps**
  - Syntaxes map to this structure, as do TMQL and TMCL
  - Also used in third-party specifications like LTM, RTM\*, TMR\*, and tolog
- **Defines the meaning of topic map concepts using prose**
  - Defines “subject”, “topic”, “scope”, “association”, ...
- **Defines their structure using the information set model**
  - Just like XML Infoset
  - Describes the kinds of things that exist in topic maps, and their properties
- **Adds constraints on the model**
  - Rules for allowed values
  - Also defines when merging happens, and how
- **Some modifications relative to XTM 1.0**

\* RDF2TM and TM2RDF specifications from Ontopia

# How TMDM works

---

- **Semantics of topic map constructs defined in prose**
- **One information item type defined for each topic map construct**
  - Complete list shown below
- **One set of properties defined for each construct**
  - Example below: all topic map objects have source locators



# TMDM metamodel

---

- **Infoset formalism used for the data model**
  - UML diagrams just for illustration
  - equality rule defined for each item type
- **Fundamental types**
  - strings, sets, and null
- **Locators**
  - strings that follow a particular syntax
  - TMDM uses URIs everywhere
- **Datatypes**
  - identified by URI
  - specify the interpretation of strings in the model
  - core datatypes: string, URI, and XML
- **Constraints**
  - prose constraints specify additional rules

## Comments

---

- **Is the datatype URI a PSI?**
  - If yes, does this mean you can reify datatypes?
  - And merge them?
  - Or, should it be a source locator so we don't have to get into this?
- **XML datatype**
  - make it clear that it has to be well-formed
  - fragment/complete document? (relationship to XML Fragment Interchange)

# Topic map

---

- **Definition**

- *"A topic map is a set of topics and associations. Its purpose is to convey information about subjects through the assignment of characteristics to topics representing those subjects. The topic map itself has no meaning or significance beyond its use as a container for the information about those subjects; in particular, the topic map does not represent anything but itself."*

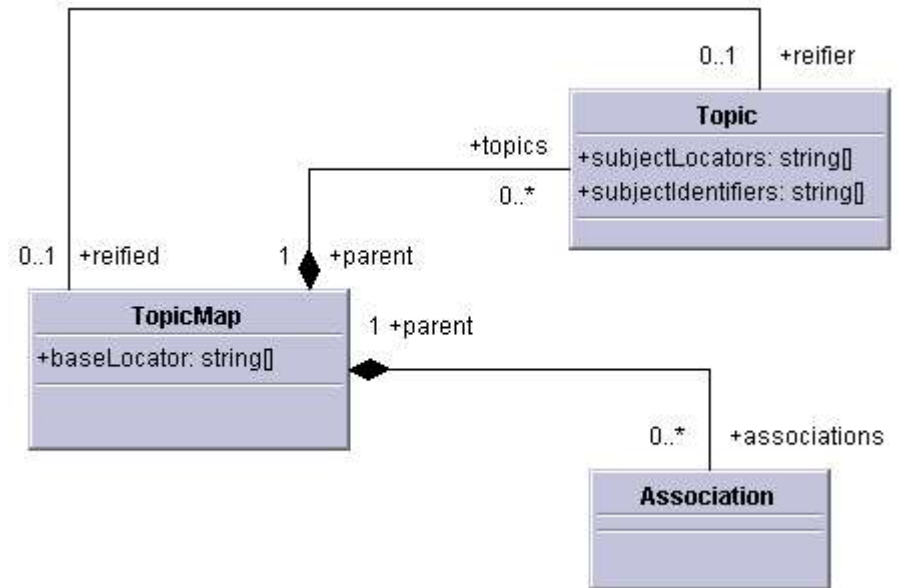
# Topic map items

---

- **Properties**

- **[topics]**: a set of topic items
- **[associations]**: a set of association items
- **[reifier]**: a topic item or null (computed property)
- **[base locator]**: a string or null
- **[source locators]**: a set of strings

- **No equality rule**



## Source locators?

---

- **These are URIs and just serve as identifiers for topic map constructs**
- **When deserializing XTM files the *id* attributes turn into source locators of the form**
  - `<full URI to XTM file>#<id>`
- **Loading `http://www.topicmaps.org/xtm/1.0/core.xtm` will get you a topic map item with one source locator:**
  - `http://www.topicmaps.org/xtm/1.0/core.xtm#xtm1.0-psi-core`
- **Other syntax specifications (such as LTM) do the same**
- **Source locators can be freely assigned, since they don't mean anything specific**

# Subject and topic

---

- **Subject:**
  - *"A subject can be anything whatsoever, regardless of whether it exists or has any other specific characteristics, about which anything whatsoever may be asserted by any means whatsoever. In particular, it is anything on which the creator of a topic map chooses to discourse."*
- **Topic:**
  - *"A topic is a symbol used within a topic map to represent some subject, about which the creator of the topic map wishes to make statements. Topics are used to represent subjects in order to allow statements to be made about the subjects through the assignment of characteristics to the topics that represent them."*

# Subject identification

---

- **Subject indicator**

- *"A subject indicator is an information resource that is referred to from a topic map in an attempt to unambiguously identify the subject of a topic to a human being. Any information resource can become a subject indicator by being referred to as such from within some topic map, whether or not it was intended by its publisher to be a subject indicator."*

- **Subject identifier**

- *"A subject identifier is a locator that refers to a subject indicator. Topic maps contain only subject identifiers (and not the corresponding subject indicators), and consequently it is the subject identifier that is the basis for merging; the subject indicator is ignored during merging."*

- **Subject locator**

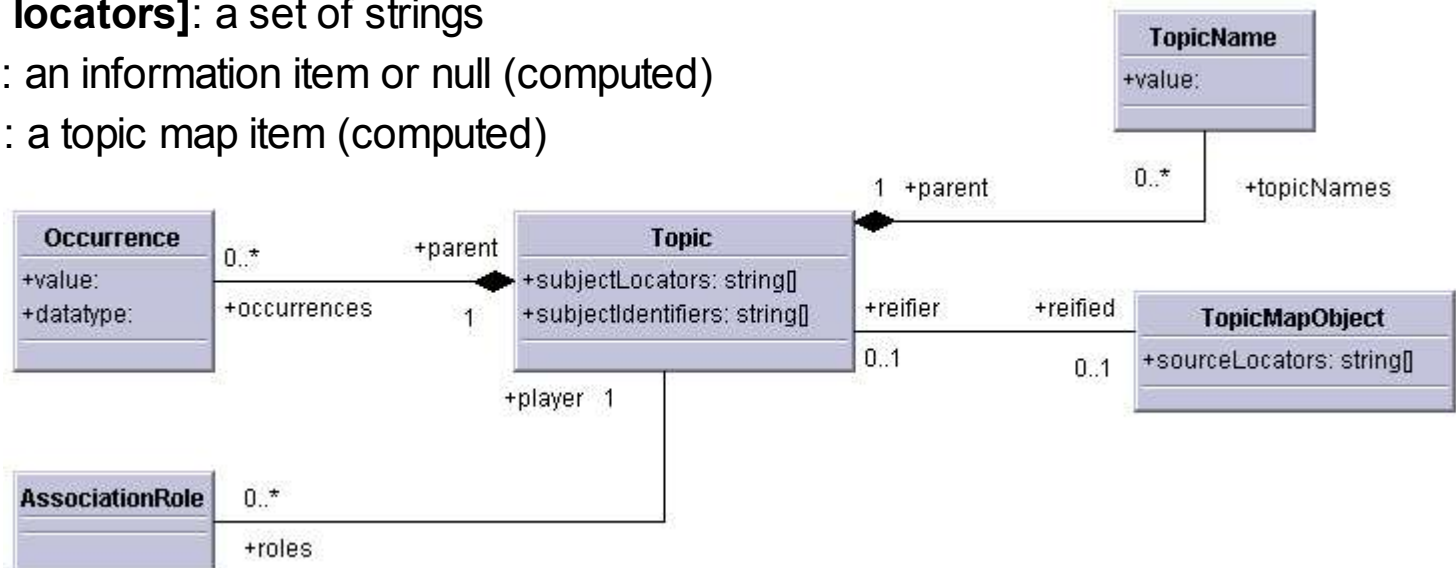
- *"A subject locator is a locator that refers to the information resource that is the subject of a topic. The topic thus represents that particular information resource; i.e. the information resource is the subject of the topic."*

# Topic items

---

- **Properties**

- **[topic names]**: a set of topic name items
- **[occurrences]**: a set of occurrence items
- **[roles played]**: a set of association role items (computed)
- **[subject identifiers]**: a set of strings
- **[subject locators]**: a set of strings
- **[source locators]**: a set of strings
- **[reified]**: an information item or null (computed)
- **[parent]**: a topic map item (computed)



## When do topics merge?

---

- **When they share a value in one of these properties:**
  - [subject identifiers]
  - [subject locators]
  - [source locators]
  - [reified]
- **When one has the same value in [source locators] as the other has in [subject identifiers]**

## Comment

---

- **The name "reification" scares people; try to make it clear that it isn't really that hard**
  - Change the name? Or just add a note? Mention "thingification"? Or just explain the etymology? "Topicification"?
- **Definition of "topic identity required" constraint shouldn't mention null**

# Topic name items

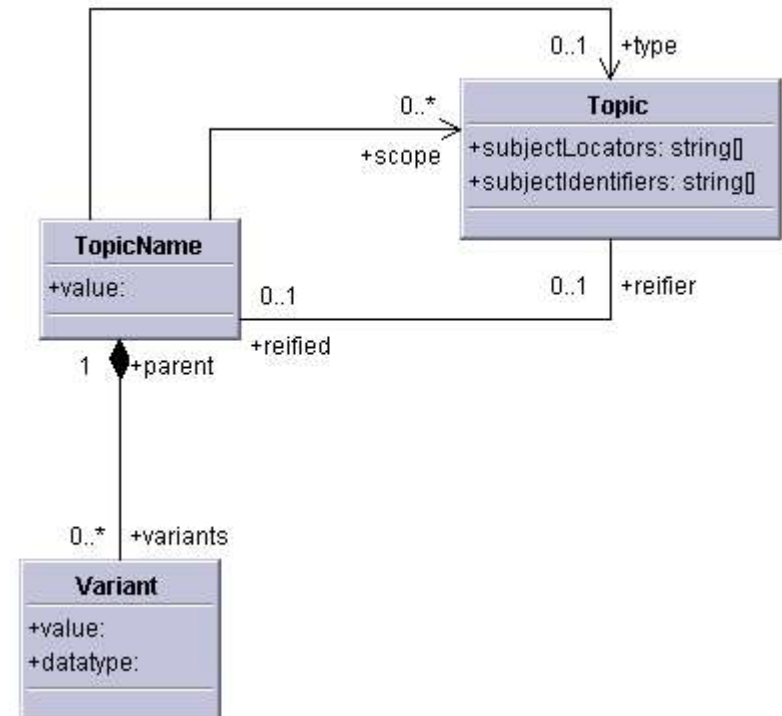
---

- **Properties**

- **[value]**: a string
- **[type]**: a topic item, or null
- **[scope]**: a set of topic items
- **[variants]**: a set of variant name items
- **[reifier]**: a topic item or null (computed)
- **[source locators]**: a set of strings
- **[parent]**: a topic item (computed)

- **Equality rule**

- [value], [type], [scope], and [parent]



## Comments

---

- **Can we lose the terms "base name" and "base form"?**
  - Can we at least expunge it from the terminology?
- **Do we really need "foo type" as a formal term?**
  - If yes, add "topic name type"
- **Why don't we allow datatypes on topic names?**
  - <insert good answer here>

# Scope

---

- **Definition**

- *"All topic characteristic assignments have a scope. The scope represents the context within which a topic characteristic assignment is valid. Outside the context represented by the scope the assignment is not known to be valid. Formally, a scope is composed of a set of topics that together define the context. That is, the topic characteristic is known to be valid only in contexts where all the subjects in the scope apply."*

## Comments

---

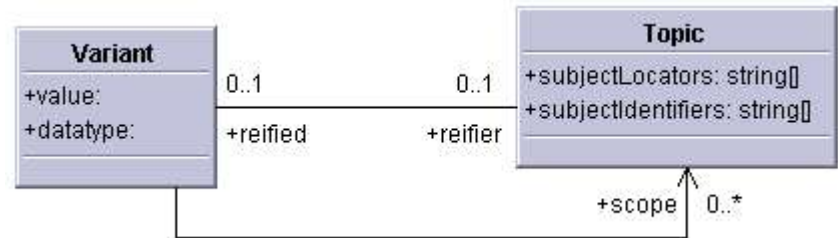
- **Best practices guide to scope needed!**
  - especially with two axes or more...

# Variant name items

---

- **Properties**

- **[value]**: a string
- **[datatype]**: a string
- **[scope]**: a set of topic items
- **[reifier]**: a topic item or null (computed)
- **[source locators]**: a set of strings
- **[parent]**: a topic name item (computed)



- **Equality rule**

- [value], [datatype], [scope], and [parent]

- **Constraint**

- *"The value of the [scope] property of each variant item must be a true superset of the value of the [scope] property of the topic name item in its [parent] property."*

## Occurrence

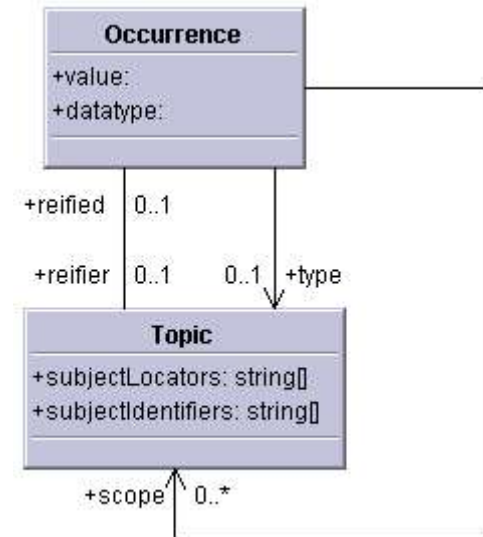
---

- ***"An occurrence is a representation of relationship between a subject and an information resource. The subject in question is that represented by the topic which contains the occurrence. The information resource may either be a value inside the topic map or an external information resource. Occurrences are essentially a specialized kind of association, where one participant in the association must be an information resource."***

# Occurrence items

---

- **Properties**
  - **[value]**: a string
  - **[datatype]**: a string
  - **[scope]**: a set of topic items
  - **[type]**: a topic item or null
  - **[reifier]**: a topic item or null (computed)
  - **[source locators]**: a set of strings
  - **[parent]**: a topic item (computed)
  
- **Equality rule**
  - [value], [datatype], [scope], [type], and [parent]



# Associations and roles

---

- **Association**
  - *"An association is a representation of a relationship between one or more subjects."*
- **Association role**
  - *"An association role is a representation of the involvement of a subject in a relationship represented by an association."*

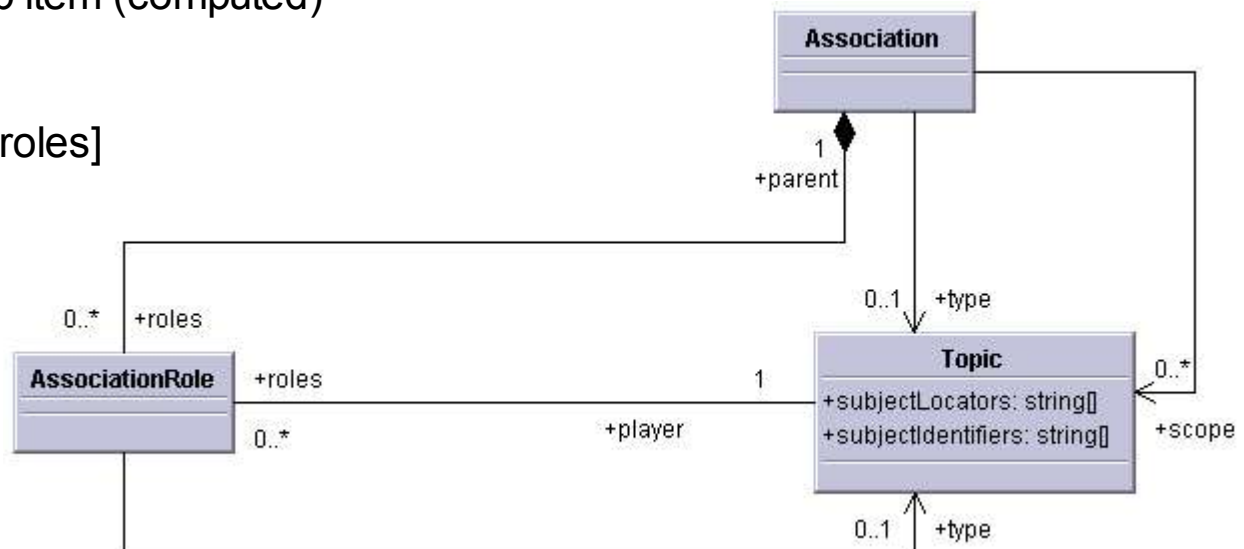
# Association items

- **Properties**

- **[type]**: a topic item or null
- **[scope]**: a set of topic items
- **[roles]**: a non-empty set of association role items
- **[reifier]**: a topic item or null (computed)
- **[source locators]**: a set of strings
- **[parent]**: a topic map item (computed)

- **Equality rule**

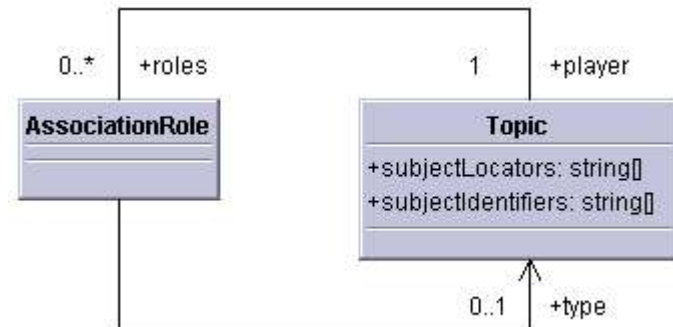
- [scope], [type], and [roles]



# Association role items

---

- **Properties**
  - **[player]**: a topic item
  - **[type]**: a topic item or null
  - **[reifier]**: a topic item or null (computed)
  - **[source locators]**: a set of strings
  - **[parent]**: an association item (computed)
- **Equality rule**
  - [type], [player], and [parent]



## Merging rules in TMDM

---

- **One merging rule defined for each information type**
  - Equality rule says which properties to compare (as for association)
  - Merging rule says how to merge two equal information items
- **Merging topics is done by**
  - creating a new topic item, whose properties contain the union of the old values,
  - then replacing all occurrences of the old items throughout the model with the new one
- **Merging is triggered by changes**
  - changes effectively defined by other specifications

## Comments

---

- Make sure it's explicit that merging is *within* topic maps, not across

## Published subjects

---

- ***"A published subject indicator is a subject indicator that is published and maintained at an advertised location for the purposes of supporting topic map interchange and mergeability."***
- ***"A published subject is any subject for which there exists at least one published subject indicator."***
- ***"A published subject identifier is the subject identifier of a published subject indicator."***
- **TMDM defines a number of these**

## Published subjects in TMDM

---

- **type-instance**
  - binary association type with two role types ("type" and "instance")
  - equivalent to <instanceOf> on topics in XTM
- **supertype-subtype**
  - binary association type with two role types ("supertype" and "subtype")
  - used to define class hierarchies
- **"sort" and "display"**
  - used to scope variant names to create sort names and display names
  - legacy from HyTM; "sort" still widely used
- **unique-characteristic**
  - a type of (topic name | occurrence | association) type which is unique
  - topics having the same name for instances of instances of these must merge
- **Topic map constructs**
  - one PSI for each information item type (except topic map and topic)

## Comments

---

- **Why is unique-characteristic part of TMDM and not outside it, for example in TMCL?**
- **Consider again whether to lose section 7.6 (or at least come up with a good reason for why it has to stay)**
- **Need a good way to represent order in topic maps**
  - <get requirements from Arnar on this>
- **Reification too heavyweight, more light-weight way to attach properties to non-topic objects needed**
- **More explanation, please, of the example at the end of 7.3**

## New things in TMDM compared to XTM 1.0

---

- **Datatypes for values**
  - Embedded XML one of the datatypes
- **Typed topic names**
- **Multiple subject locators for a single topic**
- **Defined semantics for type-instance & supertype-subtype**
- **Unique-characteristic PSI**
  - Instances of this PSI (names, occurrences, and associations) used for merging
- **Topic naming constraint gone forever**
  - (The unique-characteristic PSI is its replacement)

---

## XTM 1.1



*Overview*

*Changes from XTM 1.0*

*Some issues*

## XTM 1.1

---

- **Updated XML syntax for topic maps**
- **Backwards-compatible with XTM 1.0, but makes some extensions**
- **Provides a DTD, a RELAX-NG schema, and a W3C XML Schema**
- **The RELAX-NG schema is the normative one**
- **XTM 1.1 documents are valid if they**
  - are well-formed XML,
  - conform to XML Namespaces,
  - are valid according to the schema,
  - can be deserialized to a valid TMDM instance

## <topicMap>

---

- **Schema**

```
topicMap = element topicMap { id?, xmlbase?, version,  
                             (topic | association | mergeMap)* }
```

```
xmlbase = attribute xml:base { xsd:anyURI }
```

```
version = attribute version { "1.1" }
```

- **TMDM**

- [topics] and [associations] should be obvious
- [base locator] is set to the URI of the XTM document, if known
- the *id* if given creates a string in [source locators]
- the *xml:base* if given only changes the URIs in the document
- the *version* has no effect on the TMDM

## <topic>

---

- **Schema**

```
topic = element topic { id, instanceOf*, subjectIdentity?,  
                        (baseName | occurrence)* }
```

- **TMDM**

- a topic item is created
- the *id* creates a source locator
- other properties covered on following slides

## Comments

---

- **Why does the *id* have to be there?**
  - this causes problems in practice

## <instanceOf> inside <topic>

---

- **Schema**

```
instanceOf = element instanceOf { id?, topic-reference }
```

- **TMDM**

- creates an association using the type-instance PSI
- type is the referenced topic, instance is the parent topic
- the *id* is ignored

## Comments

---

- Why doesn't the *id* turn into a source locator on the association?

# Understanding topic references

---

- **<topicRef xlink:href="..." />**
  - if points to an external file: file is loaded, resulting TM merged into current TM
  - contents of *xlink:href* turned into absolute URI
  - topic with URI in [source locators] returned
  - if none, topic is created and source locator added
  
- **<subjectIndicatorRef xlink:href="..." />**
  - contents of *xlink:href* turned into absolute URI
  - topic with URI in [subject identifiers] returned
  - if none, topic is created and subject identifier added
  
- **<resourceRef xlink:href="..." />**
  - contents of *xlink:href* turned into absolute URI
  - topic with URI in [subject locators] returned
  - if none, topic is created and subject locator added

## Comments

---

- Is it really *required* to read in the external resource when it is referenced with `<topicRef>`?
- What about `<mergeMap>`?

## <subjectIdentity>

---

- **Schema**

```
subjectIdentity = element subjectIdentity { id?, topic-  
reference* }
```

- **TMDM**

- <topicRef> child elements turn into [source locators] values
- <subjectIndicatorRef> child elements turn into [subject identifier] values
- <resourceRef> child elements turn into [subject locators] values
- the *id* is ignored

## <baseName>

---

- **Schema**

```
baseName = element baseName { id?, instanceOf?, scope?,  
                               baseNameString, variant* }
```

- **TMDM**

- creates a topic name item
- contents of <baseNameString> go into [value] property
- topics referenced inside <scope> go into [scope] property
- topic referenced inside <instanceOf> goes into [type] property
- *id* goes into [source locators]

## <variant>

---

- **Schema**

```
variant = element variant { id?, parameters, variantName?,  
    variant* }
```

- **TMDM**

- creates a variant name item
- *id* becomes a source locator
- topics referenced within <parameters> added to [scope]
- [scope] of parent (topic name | variant name) added to this variant name
- the <variantName> element contains the actual variant information

# Comments

---

- **Lose nesting of <variant> elements?**
  - Complex to implement
  - No need for it, rarely used
  - Could we at least deprecate it so we can lose it next time?

## <variantName>

---

- **Schema**

```
variantName =
  element variantName { id?, (resourceRef | resourceData) }
  datatype = attribute datatype { xsd:anyURI }
resourceData =
  element resourceData { id?, datatype?, any-markup }
resourceRef = element resourceRef { id?, href, type? }
```

- **TMDM**

- both *id* attributes are ignored
- <resourceRef> child causes [datatype] to be URI, and [value] to be the URI
- <resourceData> child has content placed in [value] and *datatype* attribute sets [datatype], which defaults to string

## Embedded markup

---

- **Markup can now be used inside variant names and occurrences**
- **Like this:**

```
<occurrence>
  <instanceOf><topicRef xlink:href="#description"/></instanceOf>
  <resourceData
    datatype="http://www.w3.org/2001/XMLSchema#any"><p>The
    first paragraph...</p> <p>The second
    paragraph...</p></resourceData>
</occurrence>
```

## Other things to note

---

- **External occurrences can now be written with `<resourceData>`, provided the *datatype* is set to URI**
- **Datatypes not specified in TMDM can be used**
  - TMDM guarantees that the identifying URI and the string representation are preserved
  - processors which support the datatypes may do intelligent things with them

## Comments

---

- **Add note to recommend that external occurrences always use `<resourceRef>`**
- **Hyperlinking of XML element type names in HTML version**

## <occurrence>

---

- **Schema**

```
occurrence = element occurrence { id?,  
    instanceOf?, scope?, ( resourceRef | resourceData ) }
```

- **TMDM**

- creates an occurrence item
- *id* becomes a source locator
- topic referenced in <instanceOf> becomes [type]
- topics referenced in <scope> go into [scope]
- <resourceRef> and <resourceData> as for variant names

## <association>

---

- **Schema**
  - association = element association { id?, instanceOf?, scope?, member+ }
- **TMDM**
  - creates an association item
  - *id*, <instanceOf>, and <scope> as usual
  - the <member> elements create association roles in the [roles] property

## <member>

---

- **Schema**

```
member = element member { id?, roleSpec?, topic-reference* }
roleSpec = element roleSpec { id?, topic-reference }
```

- **TMDM**

- each topic-reference creates an association role item with the referenced topic in the [player] property
- if there is just one topic-reference the *id* becomes a source locator
- the topic referenced in <roleSpec> set as [type] for each role item
- if there are no topic-references a new topic item is created and set as the [player] with a unique [source locator]

## Comments

---

- Is it really a good idea to generate the source locator?

## That was it!

---

- **Questions? Complaints? Comments? Thoughts?**
- **Next meetings**
  - April 19      TMQL
  - May 3         TMCL
- **These will be more tutorial-like, and will not assume you know anything more than the basics of topic maps**