Topic Maps Self-Control

Hans Holger Rath

STEP Electronic Publishing Solutions GmbH
A division of Empolis GmbH

Germany – Hungary – Poland – UK – USA

Web: www.topicmaps.com
eMail: consulting@step.de
Sorry!

- No TM introduction
- No extensive motivation of concepts covered in this talk

But: Extreme Topic Maps
Quine's criterion:

What is there?
Everything!
Overview

- Ontologies and TM templates
  - Purpose, concept, application
- Type hierarchies
  - Super-subclassing of types
- Association properties
  - Transitivity
- Inference rules
  - Deducing implicit knowledge
- Consistency checking
  - Rule-based constraints
Ontologies and TM Templates
John F. Sowa:

"Ontology defines the kinds of things that exist in the application domain."
The ontology TM

<table>
<thead>
<tr>
<th>Topic types</th>
<th>Occurrence role types</th>
<th>Association types</th>
</tr>
</thead>
<tbody>
<tr>
<td>city map</td>
<td>city</td>
<td>is in</td>
</tr>
<tr>
<td>article</td>
<td>video</td>
<td>takes place in</td>
</tr>
<tr>
<td>call f. papers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Others:
Assoc. role types, themes
The ontology TM

Topic types
- country
- province
- city
- conference

Occurrence role types
- is in
- takes place in
- call for papers

Association types
- others: assoc. role types, themes

These are all topics!
The solution

- Term "TM template" informally defined by ISO working group
- A topic map template is a topic map
- Consists of all constructs which have a **declarative meaning** for a map
  - Topics which are candidates for themes and types
  - Inference rules (see later)
  - Constraints (see later)
The technique

- Define Public Subject Identifiers (PSIs) for the basic typing/theming topic types
  - theme
  - topic type
  - occurrence role type
  - association type
  - association role type
  - facet type
  - facet value type
The technique (real map)

<topic id="canada" types="country">
  <topname>
    <basename>Canada</basename>
  </topname>
</topic>

<topic id="montreal" types="city">
  <topname>
    <basename>Montréal</basename>
  </topname>
</topic>
The technique (template)

```xml
<topic id="country">
  <topname>
    <basename>countr</basename>
  </topname>
</topic>

<topic id="city">
  <topname>
    <basename>city</basename>
  </topname>
</topic>
```
The technique (template)

```xml
<topic id="country" types="tt">
  <topname>
    <basename>country</basename>
  </topname>
</topic>

<topic id="city" types="tt">
  <topname>
    <basename>city</basename>
  </topname>
</topic>
```
The technique (template)

```xml
<topic id="tt"
    identity="http://www.topicmaps.com/xtm/1.0/psi/topic-type">
  <topname>
    <basename>topic type</basename>
  </topname>
</topic>
```
Some remarks

- PSI should be somehow standardized (ISO, XTM) and publicly registered (ISO, OASIS)
- A typing topic type could be candidate for more than one basic type (e.g. topic type and theme)
- Naming of basic type topics is up to the application, only PSIs are fix
Type Hierarchies
The challenge

- ISO 13250 defines type-instance relationship with its "type" concept.
- Knowledge representation and advanced inferencing techniques require type hierarchies defined by supertype-subtype relationship.
The challenge

- The difference between type-instance and supertype-subtype relationships:
  - Graham is-of-type human
  - Human is-of-type species
  - Human is-subtype-of mammal

- Implies: Graham is-of-type mammal
- Implies not: Graham is-of-type species

- Supertype-subtype relationship is transitive, type-instance is not
The technique

- Define **supertype-subtype association type** and supertype and subtype association role types
- Define according PSIs:
  - .../psi/association-type/supertype-subtype
  - .../psi/association-role-type/supertype
  - .../psi/association-role-type/subtype
The technique

- Define topics for assoc. type, assoc. role types which use these PSIs
- Assign transitivity property to this association type (see later)
- Connect typing topics by this association and assign roles accordingly
The technique

- Example:

```xml
<assoc type="supertype-subtype">
  <assocrl type="supertype">mammal</assocrl>
  <assocrl type="subtype">human</assocrl>
</assoc>
```
Some remarks

- ISO will add **supertypes** attr. to topic link as shorthand for association

- The class-instance relationship expressed by the "types" attribute will also be defined as predefined association. Why?
  - Symmetry reasons
  - Attributes are shortcuts for assocs which could carry scopes
  - Cannot delete attribute from standard
Association Properties
Theoretical backgrounds

- Knowledge representation, AI, mathematics, linguistics, philosophy
- Association properties:
  - Reflexive
  - Symmetric
  - Transitive
  - Anti-reflexive
  - Anti-symmetric
- Enable automatic derivation of implicit knowledge from the map (inferencing mechanisms)
The technique

- The association type topic:

```xml
<topic id="is-in" types="at">
    <topname>
        <basename>is in</basename>
    </topname>
</topic>
```
The technique

- Assign property by facet:

```xml
<topic id="is-in" types="at">
  <topname>
    <basename>is in</basename>
  </topname>
</topic>

<facet type="association-property">
  <fvalue type="transitive">is-in</fvalue>
</facet>
```
The technique

- Assign property by facet:

```xml
<topic id="is-in" types="at">
  <topname>
    <basename>is in</basename>
  </topname>
</topic>

<facet type="association-property">
  <fvalue type="transitive">is-in</fvalue>
</facet>
```
Define facet type topics using PSI as identifier:

```xml
<topic id="association-property" types="ft" identity="http://www.topicmaps.com/xtm/1.0/psi/facet-type/property">
  <topname>
    <basename>association property</basename>
  </topname>
</topic>
```
The technique

- Define facet value type topics using PSI as identifier:

```xml
<topic id="transitive"
  types="fvt"
  identity="http://www.topicmaps.com/xtm/1.0/psi/facet-value-type/transitive">
  <topname>
    <basename>transitive</basename>
  </topname>
</topic>
```
Some remarks

- Other assoc props could be useful hints when serializing TMs as natural language or when validating TMs.
- Transitivity could be used in queries to increase or decrease the result list (e.g. up or down in type hierarchy).
Consistency Checking
The needs

- Manual checking of large maps is impossible but validation is a requirement
- TM software should validate during design and creation
  - Permanently or on demand
  - Like structure validation in SGML/XML editors/parsers
- Constraints control validation process
The techniques

- Set of topic, occurrence, and association "patterns" declared in the template
- Programming language using an API of the topic map editor/engine
- API gives more freedom, but for the price of rather big effort
- Patterns fulfill the 80/20 rule and might be sufficient for most applications
The techniques – constraint pattern

- Rule-based "patterns" for topics, occurrences, and associations
- Patterns "declare" the possible parameters and their combinations
- "Wildcard" for "any topic"
- Theme signalling if a topic must participate (playing the specified role) in an association
An example

<topic id="X"
types="country" scope="schema">
<topname min="1" scope="english">
  <basename>X</basename>
</topname>
<topname max="1" scope="french">
  <basename>X</basename>
</topname>
<occurs type="map" min="1"></occurs>
<occurs type="description" scope="english"></occurs>
<occurs type="description" scope="french"></occurs>
</topic>
Another example

```xml
<assoc type="is-in" scope="schema">
  <assocrl type="container">city</assocrl>
  <assocrl type="containee">street</assocrl>
</assoc>

<assoc type="born-in" scope="schema topic-assocrole-requirement">
  <assocrl type="person">person</assocrl>
  <assocrl type="place">place</assocrl>
</assoc>
```
Some remarks

- Constraint patterns are simple but powerful
- Smart integration of validating API functions has to be investigated (has to be checked if SAF idea could help)
Inference Rules
Background

- Type hierarchies and of transitivity allow powerful inferencing of knowledge not coded in the topic map.
- But a map may contain further knowledge which could be inferred.
- Inference rules define on the ontology level how to derive implicit knowledge.
Example

If $\text{topic1}$ is a sibling of $\text{topic2}$ and $\text{topic1}$ is a male

then $\text{topic1}$ is a brother

(Eric Freese, XML Europe 2000, Paris)
Rule components

- "if <condition> then <inference>" defines the inference rule
- "$topic1" and "$topic2" are variables which have to be instantiated when the rule is evaluated
- "is a sibling of" and "is a male" are the assoc. types in question
- "is a brother" is the inferred assoc. type
Rule components – inference rule

- Association with predefined type in predefined scope:

  ```xml
  <assoc
type="inference-rule"
scope="ir-schema">
  ...
  </assoc>
  ```
Rule components – variables

- Topic with predefined type:

```xml
<topic
  id="ir-topic-A-PERSON"
  types="ir-topvar"
  scope="ir-schema">
  <topname><basename>A PERSON</basename></topname>
</topic>
```
Rule components – assoc types

- Predefined association role types with condition number:

  ```xml
  <assocrl type="ir-cond-t1" no="2">ir-topic-A-PERSON</assocrl>
  <assocrl type="ir-cond-art1" no="2">instance</assocrl>
  <assocrl type="ir-cond-at" no="2">class-instance</assocrl>
  <assocrl type="ir-cond-art2" no="2">class</assocrl>
  <assocrl type="ir-cond-t2" no="2">male</assocrl>
  ```
Rule components – inferred assoc. type

- Predefined association role types:

  ```xml
  <assocrl type="ir-then-t1">ir-topic-A-PERSON</assocrl>
  <assocrl type="ir-then-art1">instance</assocrl>
  <assocrl type="ir-then-at">class-instance</assocrl>
  <assocrl type="ir-then-art2">class</assocrl>
  <assocrl type="ir-then-t2">brother</assocrl>
  ```
Shortcomings of straight forward solution

- Looks clumsy
- Restricted to binary associations
Elegant solution

- **Association gets id attribute** to be associated with/by other associations
- **Association patterns** model the association types in question
Elegant solution

...  
<assoc id="ir-male"
type="class-instance"
scope="ir-schema">
  <assocrl type="instance">
    ir-topic-A-PERSON</assocrl>
  <assocrl type="class">
    male</assocrl>
</assoc>

...
Elegant solution

<assoc type="inference-rule" scope="ir-schema">
  <assocrl type="ir-cond">
    ir-sibling</assocrl>
  <assocrl type="ir-cond">
    ir-male</assocrl>
  <assocrl type="ir-then">
    ir-brother</assocrl>
</assoc>
Some remarks

- Elegant solutions follows idea of constraints
- Defines TM representation of TMQL queries (variable topics and conditions are "select" and "from" part of TMQL)
Conclusions
Conclusions

- **Topic map templates**
  - Logical container for the “schema” part of a map
    - type/theme declarations
    - Constraints
    - Inference rules
  - Modularization and re-use
  - Standardization of templates

- **Association properties**
  - Transitivity
  - Reduce map size, minimize creation efforts, support inferencing capabilities
Conclusions cont’d

- Type hierarchies
  - Super-subclassing
  - Powerful inferencing
- Consistency checking with constraints
  - Rule-based constraints control validation process
  - Constraint patterns
- Inference rules
  - Deduce additional knowledge
  - Inference patterns
Conclusions cont’d

- PSIs define all necessary concepts (semantic)
- All presented features can be modeled with TM concepts in a very elegant way
- The TM control their own structure and content

=> Topic Maps self-control
See you at www.topicmaps.com

Questions?

</end>