Topic maps self-control

Hans Holger Rath
Director Consulting
STEP Electronic Publishing Solutions GmbH
Technologiepark Würzburg-Rimpar
Pavillon 7
D-97222 Rimpar, Germany
Phone: +49.9365.8062.0, Fax: +49.9365.8062.66
e-mail: consulting@step.de
http://www.step.de/
http://www.topicmaps.com/

Biography

Hans Holger Rath is director of STEP's Consulting department since 1998. He started at STEP in 1996 as senior consultant/project manager. Before he joined STEP he was head of the Document Computing department at ZGDV (Computer Graphics Center, Darmstadt, Germany). Hans Holger studied computer science and graduated with the doctoral thesis "Literate Specifying of Hypermedia Documents". All in all he has more than ten years experience in information process re-engineering gained from various projects with publishing houses, aircraft industry, and telecommunication industry. He is a frequent speaker and trainer at international SGML/XML events. Since 1998 he represents Germany in ISO/JTC1/SC34 – the ISO committee standardizing SGML, HyTime, DSSSL, and Topic Maps.

Keywords:
topic maps, ISO/IEC 13250:2000, topic map schema, schema, topic map template, type hierarchy, supertype/subtype association, association property, transitivity, inference, validation, constraint

Abstract

The ISO standard ISO/IEC 13250 Topic Maps offers a flexible and powerful technique for knowledge representation. The standard defines the general concepts and provides – with intention – just the necessary minimum of semantic. But knowledge representation requires more semantic – earlier publications identified "templates", "type hierarchies", "association properties", "inferences", and "constraints". All of them can be expressed as topic maps which simplifies the processing by topic map tools. The paper introduces the "self-control" of topic maps and models the listed semantics as topic maps.

1. Introduction

Earlier published papers [FRE00] [GRO00] [RAT00A] discuss the different technical issues which are necessary for a successful application of ISO/IEC 13250:2000 Topic Maps [ISO00] to the application domain *knowledge representation*. Most of these issues were brought to the ISO SC34/WG3 and to the IDEAlliance XTM working group for further investigation and – probably – adoption.

The issues are:

-- *Topic map templates*: the declarative part of a map containing topics which are type or theme candidates.

-- *Type hierarchies*: the supertype/subtype association between typing topics.

-- *Association properties*: the transitivity property of binary associations.

-- *Inference rules*: rules which define possible deduction of not explicitly coded knowledge.

-- *Constraints*: constraining conditions supporting guided editing and semantic validation of topic maps.

The sum of all listed concepts results in a *topic map schema*. The schema itself can be expressed as a topic map. Thus, the "schema" map controls the "real" map and defines the necessary semantic needed by topic map tools. The paper proposes a solution and gives examples for every conceptual part of the schema.

All concepts rely on the use of somehow standardized public subject identifiers.[1] All proposed notations will use the XML namespace naming rule using STEP's www.topicmaps.com URL – this will be changed when ISO and/or XTM define official PSI notation(s).

Additional concepts like the topic maps query language (TMQL) and user profiling [KSI00] are not in the scope of this paper, but both could also be expressed as topic maps.

2. Topic map templates

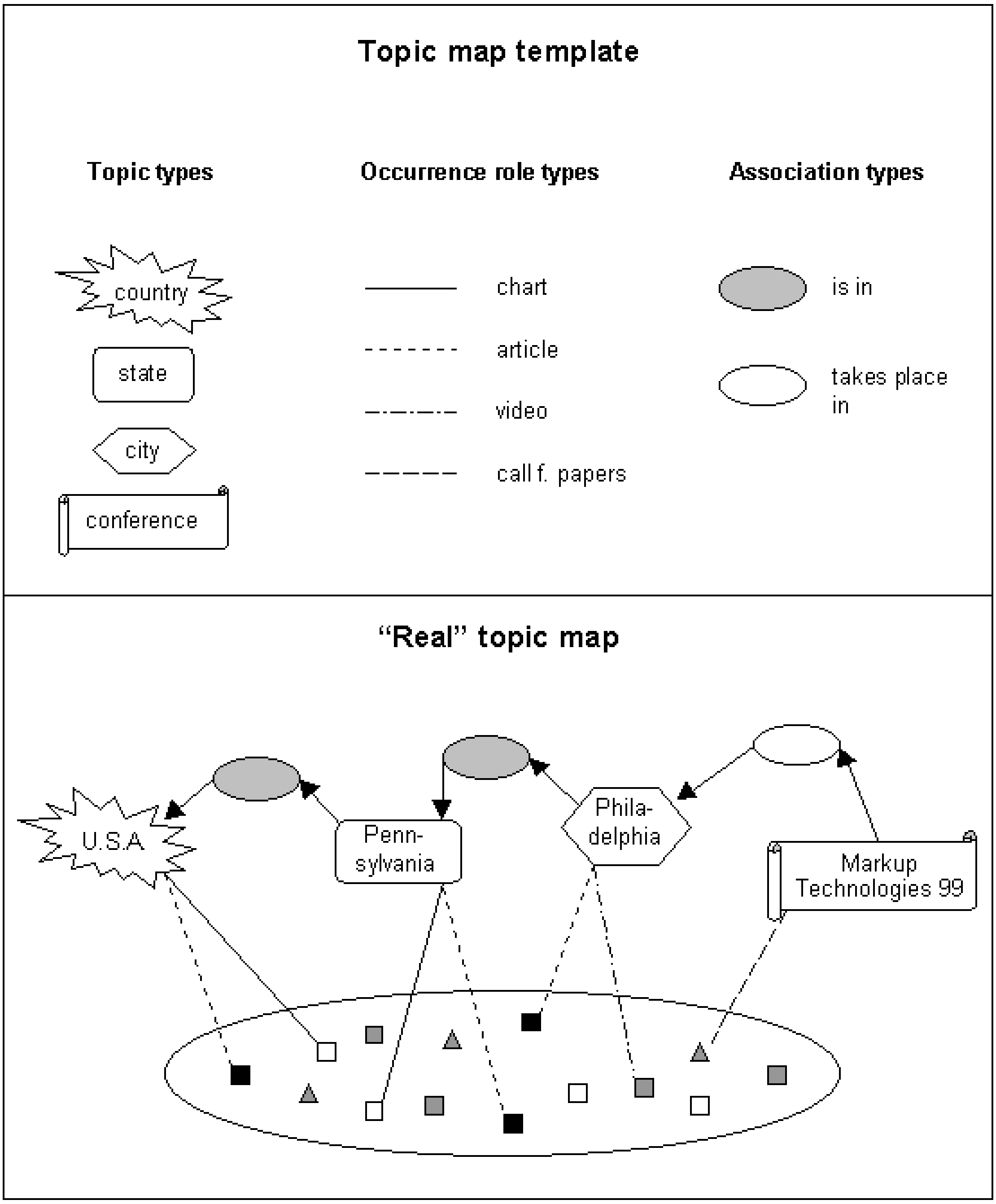
All “objects” declaring a topic map ontology[2] are topics, namely themes, topic types, occurrence role types, association types, association role types, facet types, and facet value types.

But the standard does not provide a name or definition for the list of declarative “objects” of a map and this can lead to some confusion: Users often mix up “declarative” topics and “regular” topics during discussions. In addition to that, the different tasks of topic map design, creation, and maintenance are hard to distinguish and to separate. The same is true for user access rights: As long there is no distinction, different rights cannot be assigned to the different parts of the map.

A separate declarative part could also be used for defining classes of topic maps that share a common set of typing topics with predefined semantics.

The ISO working group has already responded to the need to be able to separate the declarative part of a topic map. It coined the term *topic map template* for all declarative topics of a map. At the present time this term is only “semi-official”, since the concept has not yet been refined and added to the standard.

Figure: Distinction between topic map template and “real” topic map



A template is a topic map that consists of themes and typing topics. We will see later that it also consists of consistency constraints. We define a set of PSIs to define the basic typing topic types.

Table: PSIs for *basic types*.

Basic type	PSI
theme	http://www.topicmaps.com/xtm/1.0/psi/theme
topic type	http://www.topicmaps.com/xtm/1.0/psi/topic-type
occurrence role type	http://www.topicmaps.com/xtm/1.0/psi/occurrence-role-type
association type	http://www.topicmaps.com/xtm/1.0/psi/association-type
association role type	http://www.topicmaps.com/xtm/1.0/psi/association-role-type
facet type	http://www.topicmaps.com/xtm/1.0/psi/facet-type
facet value type	http://www.topicmaps.com/xtm/1.0/psi/facet-value-type

These basic types are declared as typing topics in the template ...

```
<topic id="th"
 identity="http://www.topicmaps.com/xtm/1.0/psi/theme">
  <topname>
    <basename>theme</basename>
  </topname>
</topic>
<topic id="tt"
 identity="http://www.topicmaps.com/xtm/1.0/psi/topic-type">
  <topname>
    <basename>topic type</basename>
  </topname>
</topic>
<topic id="ort"
 identity="http://www.topicmaps.com/xtm/1.0/psi/occurrence-role-type">
  <topname>
    <basename>occurrence role type</basename>
  </topname>
</topic>
<topic id="at"
 identity="http://www.topicmaps.com/xtm/1.0/psi/association-type">
  <topname>
    <basename>association type</basename>
  </topname>
</topic>
<topic id="art"
 identity="http://www.topicmaps.com/xtm/1.0/psi/association-role-type">
  <topname>
    <basename>association role type</basename>
  </topname>
</topic>
<topic id="ft"
 identity="http://www.topicmaps.com/xtm/1.0/psi/facet-type">
  <topname>
    <basename>facet type</basename>
  </topname>
</topic>
<topic id="fvt"
 identity="http://www.topicmaps.com/xtm/1.0/psi/facet-value-type">
  <topname>
    <basename>facet value type</basename>
  </topname>
</topic>
```

... and are used to declare the themes and typing topics of the application domain.

```
<topic id="english" types="th">
  <topname>
    <basename>english</basename>
  </topname>
</topic>
<topic id="french" types="th">
  <topname>
    <basename>french</basename>
  </topname>
</topic>
<topic id="country" types="tt">
  <topname>
    <basename>country</basename>
  </topname>
</topic>
<topic id="city" types="tt">
```

```
  <topname>
    <basename>city</basename>
  </topname>
</topic>
```

The naming of the ids and the topic names is up to the application designer. Just the PSIs are predefined. An application specific type might refer to more than one basic type (e.g. if it should be used as a topic type and as an association role type).

The “real” topic map uses the domain specific types.

```
<topic id="canada" types="country">
  <topname>
    <basename>Canada</basename>
  </topname>
</topic>
<topic id="montreal" types="city">
  <topname>
    <basename>Montréal</basename>
  </topname>
</topic>
```

3. Type hierarchies

All topics, occurrences, and associations can be seen as instances of classes (types). The classes themselves are expressed as topics. This class-instance relationship is in fact merely a syntactically privileged association type defined in the standard.

If we are looking at the class-instance relation from an object oriented view, then there is a justifiable demand for a supertype-subtype relationship as well. However, the standard explicitly declares that such a relationship has to be user-defined.

The realization of the need for type hierarchies stems from STEP's encyclopedia projects. A topic map for a lexicon contains a very large number of topics (typical orders of magnitude are hundreds of thousands or millions) and associations (even more). But most of the topic, association, and occurrence role types can be reduced to a small number of “supertypes”.

The class hierarchies become even more important when the end-user navigates or queries the map. If someone would like to know “Which pieces of music were composed by persons born in Germany that were influenced by W.A. Mozart?”, it is very likely that this information is not exactly part of the map. But with just a few topics, transitive associations, and a class hierarchy the answer can be found very easily using inference mechanisms.

Both class hierarchies and association type properties (see next section) are the basis for compact topic maps, minimized creation and maintenance efforts, and a reduction of coding errors.

The technical solution makes again use of PSIs. We need them for the supertype-subtype association, supertype association role, and subtype association role.

Table: PSIs for *supertype-subtype* association.

Predefined type	PSI
association type "supertype-subtype"	http://www.topicmaps.com/xtm/1.0/psi/association-type/supertype-subtype
association role type "supertype"	http://www.topicmaps.com/xtm/1.0/psi/association-role-type/supertype
association role type "subtype"	http://www.topicmaps.com/xtm/1.0/psi/association-role-type/subtype

The template makes use of these PSIs.

```
<topic id="supertype-subtype" types="at"
 identity="http://www.topicmaps.com/xtm/1.0/psi/association-type/supertype-subtype">
  <topname>
    <basename>supertype/subtype</basename>
  </topname>
</topic>
<topic id="supertype" types="art"
 identity="http://www.topicmaps.com/xtm/1.0/psi/association-role-type/supertype">
  <topname>
    <basename>supertype</basename>
  </topname>
</topic>
```

```
<topic id="subtype" types="art"
 identity="http://www.topicmaps.com/xtm/1.0/psi/association-role-type/subtype">
  <topname>
    <basename>subtype</basename>
  </topname>
</topic>
```

The supertype-subtype association is transitive by definition which is defined by a facet (see next section for details).

```
<facet type="association-property">
  <fvalue type="transitive">supertype-subtype</fvalue>
</facet>
```

Example in the application domain:

```
<topic id="person" types="tt">
  <topname>
    <basename>person</basename>
  </topname>
</topic>
<topic id="artist" types="tt">
  <topname>
    <basename>artist</basename>
  </topname>
</topic>
<topic id="composer" types="tt">
  <topname>
    <basename>composer</basename>
  </topname>
</topic>
<assoc type="supertype-subtype">
  <assocrl type="supertype">person</assocrl>
  <assocrl type="subtype">artist</assocrl>
</assoc>
<assoc type="supertype-subtype">
  <assocrl type="supertype">artist</assocrl>
  <assocrl type="subtype">composer</assocrl>
</assoc>
```

A `suprtyps` attributes added to the `topic` element could be a shorthand for the supertype-subtype association. The advantage of the verbose association is the possibility to assign a scope. The shorthand is easier to teach and to type.

3.1. Class instance relationship as association

For consistency reasons also the class-instance relationship modeled by the `types` attribute of the `topic` element should be modeled as an association. The technical solution makes again use of PSIs.

Table: PSIs for *class-instance* association.

Predefined type	PSI
association type "class-instance"	http://www.topicmaps.com/xtm/1.0/psi/association-type/class-instance
association role type "class"	http://www.topicmaps.com/xtm/1.0/psi/association-role-type/class
association role type "instance"	http://www.topicmaps.com/xtm/1.0/psi/association-role-type/instance

The template makes use of these PSIs.

```
<topic id="class-instance" types="at"
 identity="http://www.topicmaps.com/xtm/1.0/psi/association-type/class-instance">
  <topname>
    <basename>class/instance</basename>
  </topname>
</topic>
<topic id="class" types="art"
 identity="http://www.topicmaps.com/xtm/1.0/psi/association-role-type/class">
  <topname>
    <basename>class (type)</basename>
  </topname>
</topic>
<topic id="instance" types="art"
 identity="http://www.topicmaps.com/xtm/1.0/psi/association-role-type/instance">
  <topname>
    <basename>instance</basename>
  </topname>
</topic>
```

Example in the application domain:

```
<topic id="country">
  <topname>
    <basename>person</basename>
  </topname>
</topic>
<assoc type="class-instance">
  <assocrl type="class">tt</assocrl>
  <assocrl type="instance">country</assocrl>
</assoc>
<topic id="canada">
  <topname>
    <basename>Canada</basename>
  </topname>
</topic>
<assoc type="class-instance">
  <assocrl type="class">country</assocrl>
  <assocrl type="instance">canada</assocrl>
</assoc>
```

4. Association properties

Mathematics define the properties reflexive, symmetric, transitive, anti-reflexive, and anti-symmetric for binary relationships. Because associations can be seen as relationships the properties could be applied to associations which connect two topics. Taking a closer look on the benefits of each property shows that only transitivity is of real value. Again, PSIs define facet and facet value types to mark a transitive association.[3]

Table: PSIs for association property *transitive*.

Property	PSI
facet type "property"	http://www.topicmaps.com/xtm/1.0/psi/facet-type/property
facet value type "transitive"	http://www.topicmaps.com/xtm/1.0/psi/facet-value-type/transitive

Their use in a template:

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

Example in the application domain:

```
<topic id="is-in" types="at">
  <topname>
    <basename>is in</basename>
  </topname>
</topic>
<facet type="association-property">
  <fvalue type="transitive">is-in</fvalue>
</facet>
```

5. Inference rules

The definition of supertype-subtype relationships between types and of transitivity properties for associations already allow powerful inferencing of knowledge not coded in the topic map. But a map may contain further knowledge which could be inferred if the inference rules are specified.

5.1. Example[4]

If #topic1 is a sibling of #topic2 and #topic1 is a male then #topic1 is a brother.

This quite simple rule has to be analyzed further:

-- “if <condition> then <inference>” defines the inference rule. The “<condition>” is built by several sub-conditions connected by a logical (boolean) “and”.

-- “#topic1” and “#topic2” are variables which have to be instantiated when the rule is evaluated (= matched against the concrete topic map).
-- “is a sibling of” is an association type.
-- “is a male” is a class-instance relationship expressed by the `types` attribute or by the class-instance association.
-- “is a brother” is the inferred class-instance relationship.

5.2. First straight forward solution

Expressing this rule with topic map constructs can be done as follows:

-- The “if ... then ...” rule is coded as an association with predefined type resp. occurrence role types.
-- The variables are coded as topics of a predefined type.
-- A single condition is expressed by five different association roles – some of them are optional. The role point to either the variable topics or real topics. The real topics represent the topic types, association types, and association role types of the template.

The “brother” inference rule example:

-- Two topic variables *A PERSON* and *SIBLING* are declared in the scope *ir-schema* (inference rule schema).

```
<topic id="ir-topic-A-PERSON" types="ir-topvar" scope="ir-schema">
  <topname><basename>A PERSON</basename></topname>
</topic>
<topic id="ir-topic-SIBLING" types="ir-topvar" scope="ir-schema">
  <topname><basename>SIBLING</basename></topname>
</topic>
```

-- The association of type *inference-rule* defines the “if ... then ...” rule.

```
<assoc type="inference-rule" scope="ir-schema">
```

-- The first condition – the *no* attribute signals that the parameters belong to the first condition – requires that the variable topic *A PERSON* is associated with another variable topic *SIBLING* by an association of type *is-sibling*.

```
  <assocrl type="ir-cond-t1" no="1">ir-topic-A-PERSON</assocrl>
  <assocrl type="ir-cond-at" no="1">is-sibling</assocrl>
  <assocrl type="ir-cond-t2" no="1">ir-topic-SIBLING</assocrl>
```

-- The second condition requires that the variable topic *A PERSON* is associated with the topic *male* by an association of type *class-instance*.[5] The variable topic has to play the association role *instance* and the *male* topic has to play the role *class*.

```
  <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>
```

-- The “then” part of the rule says that – if the conditions hold in the “real” map – a new association of type *class-instance* can be asserted between all the instantiations of the variable topic *A PERSON* (playing role *instance*) and the topic *brother* (playing role *class).*

```
  <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>
```

-- The end of the inference rule association.

```
</assoc>
```

The presented solution looks a little bit clumsy because of all the different association role types and the number attributes. Furthermore, it is restricted to binary associations.

5.3. Elegant solution

A more elegant solution requires an extension of the standard: associations should be treated as topics. Which means that they would have ids and could be associated. This extension is already discussed in the topic map working groups.

With the extension the inference rules could point to association pattern which represent the condition (see also below the constraint pattern mechanism).

The elegant “brother” inference rule example:

-- Definition of topic variables.

```
<topic id="ir-topic-A-PERSON" types="ir-topvar" scope="ir-schema">
  <topname><basename>A PERSON</basename></topname>
</topic>
<topic id="ir-topic-SIBLING" types="ir-topvar" scope="ir-schema">
  <topname><basename>SIBLING</basename></topname>
</topic>
```

-- The *is-sibling* association pattern uses the topic variables. The *any* association role type represents “any” role type.

```
<assoc id="ir-sibling" type="is-sibling" scope="ir-schema">
  <assocrl type="any">ir-topic-A-PERSON</assocrl>
  <assocrl type="any">ir-topic-SIBLING</assocrl>
</assoc>
```

-- The *class-instance* association pattern relates the topic variable *A PERSON* with the *male* topic. Both play their appropriate roles in the association.

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

-- This *class-instance* association pattern relates the topic variable *A PERSON* with the *brother* topic. Both play their appropriate roles in the association.

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

-- The association of type *inference-rule* refers to the three other associations and assigns them the appropriate roles. The sibling and male association patterns become conditions – implicitly connected by boolean “and”[6] – and the brother association pattern becomes the “then” of the inference rule.

```
<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>
```

The following table lists all necessary PSIs.

Table: PSIs for inference rules.

Property	PSI
topic map "object" is part of inference rule schema[7]	http://www.topicmaps.com/xtm/1.0/psi/theme/inference-rule-schema
topic type "topic variable"	http://www.topicmaps.com/xtm/1.0/psi/topic-type/inference-variable
association type "inference rule"	http://www.topicmaps.com/xtm/1.0/psi/association-type/inference-rule
association role type "inference condition"	http://www.topicmaps.com/xtm/1.0/psi/association-role-type/inference-condition
association role type "inference statement"	http://www.topicmaps.com/xtm/1.0/psi/association-role-type/inference-statement

The use of the PSIs in the template:

```
<topic id="ir-schema" types="th"
 identity="http://www.topicmaps.com/xtm/1.0/psi/theme/inference-rule-schema">
  <topname>
    <basename>inference rule schema</basename>
  </topname>
</topic>
<topic id="ir-topvar" types="tt"
```

```
 identity="http://www.topicmaps.com/xtm/1.0/psi/topic-type/inference-variable">
  <topname>
    <basename>inference topic variable</basename>
  </topname>
</topic>
<topic id="inference-rule" types="at"
 identity="http://www.topicmaps.com/xtm/1.0/psi/association-type/inference-rule">
  <topname>
    <basename>inference rule</basename>
  </topname>
</topic>
<topic id="ir-cond" types="art"
 identity="http://www.topicmaps.com/xtm/1.0/psi/association-role-type/inference-condition">
  <topname>
    <basename>inference condition</basename>
  </topname>
</topic>
<topic id="ir-then" types="art"
 identity="http://www.topicmaps.com/xtm/1.0/psi/association-role-type/inference-statement">
  <topname>
    <basename>inference statement</basename>
  </topname>
</topic>
```

5.4. Outlook

The inference condition roles could also refer to topic pattern (see below). This would bring the expressive power of constraints to the inference rules. Having this, it will be a simple step towards the topic maps query language TMQL, because the topic variables could be seen as the “select” part and the conditions can be seen as the “from” part of TMQL. Thus, only the “where” part has to be modeled.

6. Consistency constraints

The standard has almost nothing to say on the subject of validation and consistency. The “Conformance” section of the standard focuses on the understanding of the defined constructs, the interchange syntax, and import/export of topic maps.

Both the designer and the editor of topic maps need system support when designing and creating a map which will consist of millions of topics and associations. The question of the consistency of the map becomes a key issue, because it is nearly impossible to check a map of that size manually. For that reason we need concepts to declare consistency constraints and to validate that those constraints have been obeyed. [RAT00A] [GRO00]

Consequently a separate schema is needed which contains all the information necessary for the validation process. We call this construct *consistency constraints* or just *constraints*. The validation is the task of the topic map development environment (e.g. a editor or an editorial system). It should be performed permanently or on demand – like structure validation in an SGML/XML editor.

The constraints are either a set of topic, occurrence, and association “patterns” declared in the template or implemented with a programming language using an API library of the topic map editor/engine. Latter one gives more freedom, but for the price of rather big effort. The first one fulfills the 80/20 rule and might be sufficient for most applications.

The constraints are defined rule-based as “patterns” for topics, occurrences, and associations. These “patterns” “declare” the possible parameters and their combinations. The patterns are defined as topics and associations. A predefined theme “schema” which is used in the `scope` attribute signals that these topics and associations have a special meaning – they are constraints for topics/associations of the given type.

A predefined topic will be used if the pattern needs a wildcard for topics (*any topic*). A predefined theme is assigned to the association as scope if some topics must participate (playing the specified role) in the association.

These PSIs define the constraint schema theme, the any topic, and the required association role theme.

Table: PSIs for *constraints*.

Description	PSD
topic map "object" is part of constraint schema	http://www.topicmaps.com/xtm/1.0/psi/theme/schema
the *any* topic	http://www.topicmaps.com/xtm/1.0/psi/topic/any
topic has to be used in association	http://www.topicmaps.com/xtm/1.0/psi/theme/topic-assocrl-requirement

6.1. Use of PSIs in template

```
<topic id="schema" types="th"
 identity="http://www.topicmaps.com/xtm/1.0/psi/theme/schema">
  <topname>
    <basename>constraint schema</basename>
  </topname>
</topic>
<topic id="any"
 identity="http://www.topicmaps.com/xtm/1.0/psi/topic/any">
  <topname>
    <basename>any topic</basename>
  </topname>
</topic>
<topic id="topic-assocrole-requirement"
 identity="http://www.topicmaps.com/xtm/1.0/psi/theme/topic-assocrl-requirement">
  <topname>
    <basename>topic association role requirement</basename>
  </topname>
</topic>
```

6.2. Topic type examples

An example of a pattern that constraints the topic type *country*:

```
<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>
```

Explanation of the constraint for topics of type country:

-- A capital *X* means *any value*.

⇒ Any value for attribute *id* and element *basename*.

-- The *min* attribute means that there has to be a minimum number of instances of the element in the topic map – but more than that number are allowed.

⇒ 1–*n* elements *topname* with scope *english* and 1–*n* occurrences of type *map*.

-- The *max* attribute means that there could be a maximum number of instances of the element in the topic map – but less than that number are allowed.

⇒ 0–1 element *topname* with scope *french*.

-- If no *min* and no *max* attribute is set the exact number has to be in the topic map.

⇒ One occurrence of type *description* in scope *english* and one occurrence of type *description* in scope *french*.

6.3. Association type examples

There is also a pattern for association types which controls the scope, the combination of valid association role types using *min*/*max* attributes, and the valid topic types for every role.

```
<assoc type="born-in" scope="schema biography">
  <assocrl type="person">person</assocrl>
  <assocrl type="place">geo-object</assocrl>
</assoc>
```

Explanation of the constraint for associations of type born-in:

-- A theme besides “schema” assigned to the `scope` attribute signals that the association has to be of this scope.

⇒ All *born-in* associations have to be in the *biography* scope.

-- The listed association roles are mandatory – because no *min* attribute is specified.

⇒ All *born-in* associations have to have the two association roles *person* and *place* referring to topics of type *person* resp. *geo-object*.

```
<assoc type="is-in" scope="schema">
  <assocrl type="container">country</assocrl>
  <assocrl type="containee">state</assocrl>
</assoc>
<assoc type="is-in" scope="schema">
  <assocrl type="container">country</assocrl>
  <assocrl type="containee">county</assocrl>
</assoc>
<assoc type="is-in" scope="schema">
  <assocrl type="container">country</assocrl>
  <assocrl type="containee">city</assocrl>
</assoc>
<assoc type="is-in" scope="schema">
  <assocrl type="container">state</assocrl>
  <assocrl type="containee">county</assocrl>
</assoc>
<assoc type="is-in" scope="schema">
  <assocrl type="container">state</assocrl>
  <assocrl type="containee">city</assocrl>
</assoc>
<assoc type="is-in" scope="schema">
  <assocrl type="container">county</assocrl>
  <assocrl type="containee">city</assocrl>
</assoc>
<assoc type="is-in" scope="schema topic-assocrole-requirement">
  <assocrl type="any">any</assocrl>
  <assocrl type="containee">city</assocrl>
</assoc>
```

Explanation of the constraint for associations of type is-in:

-- Only the listed association role / topic type combinations are valid.

⇒ The container/containee roles can be played by these topic type combinations: country/state, country/county, country/city, state/county, state/city, county/city.

-- The schema *topic-assocrole-requirement* signals that every topic of explicitly given type has to play the listed role in at least one association of the given type.

⇒ Every topic of type *city* has to play the role *containee* in at least one association of type *is-in*.

```
<assoc type="meeting" scope="schema">
  <assocrl type="location">geo-object</assocrl>
  <assocrl type="chair">person</assocrl>
  <assocrl type="participant" min="2" max="20">person</assocrl>
</assoc>
<assoc type="conference" scope="schema">
  <assocrl type="location">geo-object</assocrl>
  <assocrl type="chair">person</assocrl>
  <assocrl type="participant" min="21">person</assocrl>
</assoc>
```

Explanation of the constraint for associations of types meeting and conference:

-- The *min* attribute means that there has to be a minimum number of association roles *location* in the topic map – but more than that number are allowed.

The *max* attribute means that there could be a maximum number of association roles *participant* in the topic map – but less than that number are allowed.

⇒ A *meeting* consists of at least 2 participants, but the maximum number of participants is limited to 20 – otherwise it would be a *conference* which has to have at least 21 participants.

6.4. Constraints and type hierarchies

The defined constraints are automatically valid for all subtypes of the topic type or the association type. Subtypes of the defined association role types and topic types playing that role are automatically valid as well.

The declaration of type hierarchies (e.g. type *painter* is subtype of *artist* is subtype of *person*) will simplify the declaration of constraints. Declaring the constraint for a general supertype automatically declares the same constraint for all its subtypes.

An example is Vincent van Gogh who is probably a topic of types `painter` but it is valid in the `born-in` association to play the role `person`.

7. Conclusions

Topic maps are a powerful concepts to define intelligent link networks over continuously growing information pools. Real-world topic maps will consist of a large number of objects which require validation to assure the quality of the map. A couple of concepts help with the QA or make implicitly coded knowledge explicit: topic map templates define the ontology of the application domain, type hierarchies express the supertype/subtype relationship, association properties assign transitivity to binary associations, inference rules declare how to derive implicit knowledge, and constraints are the validation rules. The sum of the concepts describe the *topic map schema*.

The paper showed that the topic map schema can be modeled as a topic map – some of the concepts require minor extensions of the element types defined by the ISO standard. Using the topic maps paradigm also for the definition of the listed control structures allows self-control of topic maps and simple handling by topic map tools.

Public subject identifies (PSIs) distinguish the schema objects from the objects of a “regular” topic map. It is the task of the ISO resp. topicmaps.org working group to standardize the PSIs.

8. Outlook

The presented concepts are the foundation for a successful application of topic maps in knowledge representation and knowledge organization. Text retrieval in semantic networks is a well-known technology which is already implemented in various products and projects in different application domains [LEBABUWE98] .

Semantic networks are a subset of topic maps [FRE00] . Thus it is obvious that text retrieval could be based on topic maps as well. Which provide a standardized format for the knowledge representation and improve semantic networks by some valuable features – like scopes, n-ary associations, and merge functionality. Existing algorithms implemented by the text retrieval systems could also be used to make the use of topic maps more valuable. These are:

-- Existing text analysis software could be used to (semi-) automatically generate topic maps from raw text resources.

-- The same or similar text analysis software could be used to translate natural language search queries into TMQL. The TMQL queries could be stored as small topic maps – as explained above – and used as “profiles” for further queries [KSI00] .

-- Text retrieval algorithms could be used to focus on the relevant part of the topic map instead of navigating to it. Navigation in the map starts when the relevant part is found. This means that the rich topic map navigation capabilities are connected with the powerful search capabilities of intelligent text retrieval. This combination gives the user the best of both worlds.

-- The result list of a text retrieval could be dynamically transformed into a topic map which could be visually navigated. If the user gets to the “end” of the result map he can switch back to the “full” map or can start an additional text retrieval.

This impressive list shows that topic maps should not only been seen and used as a stand-alone technology, but in combination with existing knowledge exploration and organization concepts. The synergy effects are large and an integrated system will be a big step towards easy to use and easy to implement knowledge management – in the real meaning of both words.

Acknowledgements

The author would like to thank all “topic mappers” at STEP for their input and open discussions about topic map self-control. Furthers credits go to Mario Lenze and Wolfgang Wilke of tec:inno for their valuable concepts concerning the integration of text retrieval and topic maps.

Bibliography

[Fre00]
Freese, E.: *Using Topic Maps for the representation, management & discovery of knowledge*, in: Proceedings of XML Europe 2000 Conference, GCA, Alexandria, VA, 2000.

[Gro00]
Grønmo, G.O.: *Creating semantically valid topic maps*, in: Proceedings of XML Europe 2000 Conference, GCA, Alexandria, VA, 2000.

[ISO00]
International Organization for Standardization: *ISO/IEC 13250:2000 Document description and processing languages – Topic Maps*, ISO, Geneva, 2000.

[Ksi99]
Ksiezyk, R.: *Trying not to get lost with a Topic Map*, in: Proceedings of XML Europe 99 Conference, GCA, Alexandria, VA, 1999.

[Ksi00]
Ksiezyk, R.: *Answer is just a question [of matching Topic Maps]*, in: Proceedings of XML Europe 2000 Conference, GCA, Alexandria, VA, 2000.

[LeBaBuWe98]
Lenz, M.; Bartsch-Spörtl, B.; Burkhard, H.-D.; Wess, S. (eds): *Case-based reasoning technology – foundations for applications*, Springer, 1998, ISBN 3-540-64572-1.

[Pep99a]
Pepper, S.: *Euler, Topic Maps, and Revolution*, in: Proceedings of XML Europe 99 Conference, GCA, Alexandria, VA, 1999.

[Pep99b]
Pepper, S.: *Navigating haystacks and discovering needles*, in: Sperberg-McQueen, C.M.; Usdin, B.T. (eds): *Markup Languages: Theory & Practice*, 1.4, 1999.

[RaPe99]
Rath, H.H.; Pepper, S.: *Topic maps at work*, in: Goldfarb, C.F., Prescod, P. (eds): *XML Handbook*, 2nd edition, Prentice Hall, 1999.

[Rat99a]
Rath, H.H.: *Technical Issues on Topic Maps*, in: Proceedings of MetaStructures 99 Conference, GCA, Alexandria, VA, 1999.

[Rat99b]
Rath, H.H.: *Topic Map secrets – What the standard does not tell you*, in: Proceedings of XML 99 Conference, GCA, Alexandria, VA, 1999.

[Rat00a]
Rath, H.H.: *Topic maps: templates, topology, and type hierarchies*, in: Sperberg-McQueen, C.M.; Usdin, B.T. (eds): *Markup Languages: Theory & Practice*, 2.1, 2000.

[Rat00b]
Rath, H.H.: *Making topic maps more colourful*, in: Proceedings of XML Europe 2000 Conference, GCA, Alexandria, VA, 2000.

[Sow00]
Sowa, J.F.: *Knowledge Representation – Logical, Philosophical and Computational Foundations*, Brooks/Cole, 2000, ISBN 0-534-94965-7.

[1] The term *public subject identifier* (*PSI*) was recently invented by ISO JTC1/SC34/WG3. Its main difference to a public subject descriptor (PSD) is as follows: the PSI is an identifier whereas the PSD is the description (prose text) of the object identified/addressed by the PSI. The PSI does not necessarily has to be a resolvable address (e.g. HyTime, URL/URI, XLink/XPointer), but it could be.

[2] John F. Sowa: “Ontology defines the kinds of things that exist in the application domain.” [SOW00]

[3] Assigning facets to topics is not following the intention of the standard. ISO 13250 defines facets as properties about

the information resources rather than the topics. But transitivity is a *property* of the association type. Thus, it is a natural approach to use facet to assign this property. If there is a need for further association properties, like symmetry, then they can be added easily as PSI for facet value types.

[4] This example was given by Eric Freese during his presentation at XML Europe 2000, Paris.

[5] The meaning of the *class-instance* association type is identical with the *types* attribute of the *topic* element.

[6] A boolean “or” could be modeled by further inference-rule associations. More complex boolean operator combinations could be modeled with “or” and “not” associations which combine the conditions accordingly – but this might lead to quite complex association hierarchies and would probably be better solved with a programming language (see also below constraints and programming languages).

[7] The inference rule patterns have to be distinguishable from the constraint patterns. Therefore, a separate theme is needed.