Guidelines for RDF/Topic Maps Interoperability

Abstract

The Resource Description Framework (RDF) is a model developed by the W3C for representing information about resources in the World Wide Web. Topic Maps is a model for knowledge integration developed by the ISO. This document provides guidelines for users who want to combine usage of the W3C's RDF/OWL family of specifications and the ISO's family of Topic Maps standards.

Table of contents

1. Introduction
   1.1 Background
   1.2 Purpose
   1.3 Audience
   1.4 Requirements
      1.4.1 MUST requirements
      1.4.2 SHOULD requirements

2. Guidance rules
   2.1 Things and proxies
   2.2 Assertions, types, and properties
   2.3 Identity
   2.4 Names
      2.4.1 Variants
   2.5 Occurrences
   2.6 Binary associations
   2.7 Non-binary associations
   2.8 Datatypes
   2.9 Reification
   2.10 Scope
      2.10.1 Language tags
   2.11 RDF statements

3. Authoring guidelines
   3.1 Guidelines for authors of topic maps
   3.2 Guidelines for authors of RDF documents

4. The RDFTM vocabulary
   4.1 Classes
   4.2 Properties
   4.3 Built-in guidance
   4.4 The RDFTM Language vocabulary

5. Limitations
   5.1 Non-deterministic rules
   5.2 Unsupported constructs

6. Acknowledgements
7. References

1. Introduction

1.1 Background

The Resource Description Framework (RDF) is a model developed by the W3C for representing information about resources in the World Wide Web. Topic Maps is a standard for knowledge integration developed by the ISO. The two specifications were developed in parallel during the late 1990's within their separate organizations for what at first appeared to be very different purposes. The results, however, turned out to have a lot in common and this has led to calls for their unification.

While unification has to date not been possible, a number of attempts have been made to uncover the synergies between RDF and Topic Maps and to find ways of achieving interoperability at the data level. There is now widespread recognition within the respective user communities that achieving such interoperability is a matter of some urgency. This document provides a set of Guidelines for users who want to combine usage of the W3C's RDF/OWL family of specifications and the ISO's family of Topic Maps standards.

1.2 Purpose

The purpose of this document is to present a solution to the problem of RDF/Topic Maps interoperability at the data level. It consists of guidelines that describe how to author topic maps and RDF documents in order to ensure maximum interoperability, and a set of rules for performing automated translation between RDF and Topic Maps. As the word guidelines suggests, this document describes one of several possible ways to perform translations from RDF to Topic Maps, and vice versa, that is recommended as best practice.

The primary goal of these Guidelines is to enable data to be translated from one form to the other without unacceptable loss of information or corruption of the semantics. Further goals are to be able to query the results of a translation in terms of the target model and to share vocabularies across the two paradigms.

In theory, translations can be guided or unguided. An unguided translation is one that can be performed without the presence of specific information whose purpose is to affect the result of the translation. A guided translation is one that relies on the presence of such information in order to perform a correct translation.

These Guidelines focus primarily on guided translations, since these come closest to achieving the goals described above, but it also explains how to handle situations in which guidance is incomplete or missing.

We don't do this yet. In RDF2TM, every statement that lacks guidance should become an occurrence. In TM2RDF, every association that lacks guidance should be treated as an rdftm:N-aryRelation conforming to pattern 1B.

[RDF Schema] and [OWL] are considered relevant to this work to the extent that the classes and properties they define are supportive of its goals. However, it is explicitly not a goal of the current work to enable the general use of RDF Schema and OWL with Topic Maps. Neither is it a goal to enable transformations between vocabularies, or provide a general mapping between the RDF and
Topic Maps models.

1.3 Audience

This document is aimed at anyone with an interest in the problem of RDF/Topic Maps interoperability and a willingness to acquire the necessary understanding of both formalisms. In particular it targets authors of Topic Maps and RDF vocabularies, creators of tools for translating between RDF and Topic Maps, and anyone who seeks reassurance that data can be easily reused across the two paradigms.

Readers are expected to be familiar with RDF and Topic Maps concepts (to a level corresponding to the tutorial material in [RDF Primer] and [Pepper], respectively) and with the syntaxes described in [Turtle] and [LTM]. To fully understand Section 2, the reader must also be conversant with the models described in [RDF Semantics] and [TMDM].

1.4 Requirements

This section lists the requirements upon which the guidelines contained in this document were defined.

1. Data originating in one paradigm MUST merge cleanly with data originating in the other.
2. Vocabularies MUST be reusable across the two paradigms.
3. Queries written against one model MUST be usable with data translated from the other.
4. Constructs that cannot be handled by the translation mechanism (if any) MUST be specified in the Guidelines.
5. Guidelines MUST be given to authors on how to ensure maximum interoperability.
6. The results of a translation MUST be deterministic.
7. Guidance MUST be expressed using classes and properties from the RDF, RDF-S, OWL, TM, and RDFTM vocabularies only.
8. Round-tripping SHOULD be possible.
9. It SHOULD be possible to implement the translation using event-based processing.
10. Properties and classes defined in rdf, rdfs, owl, and tm SHOULD be used where possible in order to aid and guide translations.
11. There SHOULD be just one vocabulary that covers both directions and can be expressed in either RDF or Topic Maps.
12. The translation mechanism SHOULD be capable of handling every possible construct in the source paradigm.

2. Guidance rules

This section describes how constructs in the two paradigms relate to each other and the rules for translating a topic map to RDF (TM2RDF) and vice versa (RDF2TM). It is structured by concept, starting with the most general concepts (things, proxies, assertions, etc.) and ending up with concepts that are specific to one paradigm or the other (e.g., scope and language tags).

Guidance is expressed in the form of statements that annotate ontology terms using the RDF, RDF-S, OWL, and Topic Maps vocabularies, and the RDFTM vocabulary, introduced in this document, whose specific purpose is to enable translations between RDF and Topic Maps.

Explanatory examples are given using [LTM] and [Turtle] for Topic Maps and RDF respectively.
2.1 Things and proxies

Topic Maps and RDF share the common purpose of enabling assertions (or statements) to be made about things that exist in some domain of interest. The things about which assertions are made are called subjects and resources in Topic Maps and RDF, respectively.

The term subject as used in Topic Maps should not be confused with the term as used in RDF, where it denotes one of the components of an RDF triple. Note also that the term resource is commonly used in Topic Maps in the sense of information resource, which is more restricted than the RDF usage of the term.

Subjects in Topic Maps and resources in RDF can be regarded as being fundamentally equivalent. In practice, they can both be anything at all, including abstractions, objects in the real world, documents, and more.

The things about which Topic Maps and RDF allow statements to be made are represented internally in computer systems by symbols, or proxies. In Topic Maps, those proxies are called topics; in RDF they tend to be called resources, like the things they represent. A more precise term for the proxy in RDF would be RDF node, except that the latter includes both nodes with identity (denoted by URIrefs), blank nodes, and literals. In Topic Maps, literals are strings, which are distinct from topics, so there is not a complete equivalence between topics and RDF nodes.

The close semantic equivalence between topics and resources suggests that these should be mapped to each other when translating TM2RDF or RDF2TM.

Translation Rules (normative)

TM2RDF

- A topic is mapped to an RDF node.

RDF2TM

- A URIref or blank node is mapped to a topic.
- An RDF literal is mapped to the value of a name or occurrence according to the rules given in Sections 2.4 and 2.5.

Examples (informative)

See examples in Section 2.3.

2.2 Types and properties

RDF statements are generally mapped to TM statements, and vice versa (except for non-binary associations, which are dealt with in Section 2.7). Since Topic Maps distinguishes between three kinds of statement (names, occurrences, and associations), guidance must be provided as described in Sections 2.4-2.7. Topics that have multiple identification properties result in additional RDF statements. Thus, certain RDF statements are mapped to TM identification properties.
When an RDF statement maps to a TM statement, or vice versa, the URIref of the resource maps to the subject identifier of the typing topic.

**Translation Rules (normative)**

**TM2RDF**

- Association types, occurrence types, and name types are mapped to RDF properties.
- The subject identifier of the typing topic maps to the URIref of the property.
- Special considerations applying to the handling of association role types and the types of non-binary associations are covered in Sections 2.6 and 2.7.

**RDF2TM**

- RDF properties are mapped to typing topics.
- The URIref maps to the topic's subject identifier.

**Examples (informative)**

See examples in Section 2.3.

---

### 2.3 Identity

An RDF node can have at most one URIref, whereas a topic can have multiple identifiers of different types (subject identifier, subject locator, item identifier). In order to handle this situation in the most natural way and allow for round-tripping, these Guidelines introduce the class `rdftm:InformationResource`, defined as a resource that can be represented as a sequence of bytes and can thus be identified in Topic Maps using a subject locator.

**Translation Rules (normative)**

**TM2RDF**

- When a topic has one or more subject locators, one subject locator becomes the URIref of the RDF node and the resource is asserted to be an instance of `rdftm:InformationResource`. Additional subject locators become `owl:sameAs` properties. Any subject identifiers become `rdftm:subjectIdentifier` properties.
- When a topic has one or more subject identifiers and no subject locators, one subject identifier becomes the URIref of the RDF node. Any additional subject identifiers become `rdftm:subjectIdentifier` properties.
- When a topic has neither subject identifier nor subject locator it becomes a blank node.
- Item identifiers become `rdftm:itemIdentifier` properties.

**RDF2TM**

- The URIref of a resource that is an instance of `rdftm:InformationResource` becomes a subject locator. Any `owl:sameAs` properties become additional subject locators. Any `rdftm:subjectIdentifier` properties become subject identifiers.
- The URIref of a resource that is not an instance of `rdftm:InformationResource`, becomes a subject identifier. Any additional `rdftm:subjectIdentifier` properties become subject identifiers.
- Blank nodes become topics with no subject identifier or subject locator.
- An `rdftm:itemIdentifier` property becomes an item identifier.

**Examples (informative)**

NOTE: The `rdftm:itemIdentifier` property is only shown in the examples in this section and are omitted in later examples for the sake of brevity. This section assumes that the examples are taken from a topic map whose location is `<file:/usr/topicmaps/mymap.ltm>`.

**Topic with no subject identifier or subject locator**

```
[puccini]
```

```
[ rdftm:itemIdentifier
  <file:/usr/topicmaps/mymap.ltm#puccini> .]
```

**Topic with single subject identifier (and item identifier)**

```
[puccini
  @"http://en.wikipedia.org/wiki/Puccini"]
```

```
<http://en.wikipedia.org/wiki/Puccini>
  rdftm:itemIdentifier
  <file:/usr/topicmaps/mymap.ltm#puccini> .
```

**Topic with multiple subject identifiers (and item identifier)**

```
[composed-by
  @"http://psi.ontopia.net/music/composed-by"
  @"http://www.kanzaki.com/ns/music#composer"]
```

```
<http://www.kanzaki.com/ns/music#composer>
  rdftm:subjectIdentifier
  <http://psi.ontopia.net/music/composed-by>
  rdftm:itemIdentifier
  <file:/usr/topicmaps/mymap.ltm#composed-by> .
```

**Topic with subject locator**

```
[survey
  "%http://www.w3.org/TR/rdftm-survey/"
]
```

```
<http://www.w3.org/TR/rdftm-survey/>
  rdf:type rdftm:InformationResource ;
  rdftm:itemIdentifier
  <file:/usr/topicmaps/mymap.ltm#survey> .
```

**Topic with multiple subject locators**

```
[survey
  %"http://www.w3.org/TR/rdftm-survey/"
  %"http://www.w3.org/TR/rdftm-survey"
]
```

```
<http://www.w3.org/TR/rdftm-survey/>
  rdf:type rdftm:InformationResource ;
  owl:sameAs
    <http://www.w3.org/TR/rdftm-survey> ;
  rdftm:itemIdentifier
  <file:/usr/topicmaps/mymap.ltm#survey> .
```

**Topic with subject locator and subject identifier**

```
[survey
  %"http://www.w3.org/TR/rdftm-survey/"
  @"http://www.w3.org/rdftm/#Survey"
]
```

```
<http://www.w3.org/TR/rdftm-survey/>
  rdf:type rdftm:InformationResource ;
  rdftm:subjectIdentifier
    <http://www.w3.org/rdftm/#Survey> ;
  rdftm:itemIdentifier
  <file:/usr/topicmaps/mymap.ltm#survey> .
```

## 2.4 Names

A *topic name* is the equivalent of an RDF statement that assigns a label to a resource. A name thus
maps to a single statement whose predicate is the same as the name type. In order to preserve the information that the statement has the semantics of a topic name, the property is declared to be an instance of the class `rdftm:NameProperty`. Built-in guidance (see Section 4.3) declares `rdfs:label` to be an instance of this class.

Translation Rules (normative)

**TM2RDF**

- A name maps to a statement: the topic becomes the subject, the name type becomes the predicate, and the value of the name becomes the object.
- The property becomes an instance of `rdftm:NameProperty`.

**RDF2TM**

- Statements whose properties are instances of `rdftm:NameProperty` map to names.
- The property becomes the type of the name.

Examples (informative)

NOTE: The `rdftm:itemIdentifier` property is omitted in these and all further examples.

<table>
<thead>
<tr>
<th>Topic with default name type</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{puccini = &quot;Giacomo Puccini&quot;}</code></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic with specified name type</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{puccini = foaf:name : &quot;Giacomo Puccini&quot;}</code></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic with name of type <code>rdfs:label</code> (built-in guidance, see Section 4.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{puccini = rdfs:label : &quot;Giacomo Puccini&quot;}</code></td>
</tr>
</tbody>
</table>

3.4.1 Variants

A variant name is mapped to RDF by reifying the relation resulting from mapping its base name and using `rdftm:variant`, `rdftm:value`, and `rdftm:variantScope` properties.

**TM2RDF**

- A topic name that has one or more variants results in a statement as described above in Section 2.4.
- That statement is reified using the RDFTM reification vocabulary described in Section 2.9.
- The reified statement is assigned one `rdftm:variant` property for each variant, the value of which is a blank node of type `rdftm:Variant`; this blank is assigned an `rdftm:value` property for the variant name's [value] property, and an `rdftm:scope` property for each topic in the variant name's [scope] property.

**RDF2TM**

- A blank node that has RDFTM reification properties and a `rdftm:variant` property results in a topic name with a variant. (Note: The OWL ontology for RDFTM will provide constraints for
Do we want to use SKOS as an example here? Given that SKOS has a vocabulary of its own for something that approximates to variants, the RDFTM approach does not lead to natural results. At the same time, SKOS is not rich enough to accommodate both multiple names and variants, so SKOS cannot form the basis of a more general solution.

2.5 Occurrences

An occurrence is the equivalent of an RDF statement whose value is an information resource. An occurrence thus maps to a single statement whose predicate is the same as the occurrence type. In order to preserve the information that the statement has the semantics of an occurrence, the property is declared to be an instance of the class rdftm:OccurrenceProperty.

Translation rules (normative)

TM2RDF

- An occurrence maps to a statement: the topic becomes the subject, the name type becomes the
predicate, and the value of the name becomes the object.
- The property becomes an instance of `rdftm:OccurrenceProperty`.

**RDF2TM**

- Statements whose properties are instances of `rdftm:OccurrenceProperty` map to occurrences.
- The property becomes the type of the occurrence.

**Examples (informative)**

**Internal occurrence (value is a literal)**

\[
\{\text{puccini, bio:dateOfBirth, } [\{1858-12-22\}]\} \\
\begin{align*}
_\text{a000} & \text{ bio:dateOfBirth} \ "1858-12-22" . \\
& \text{bio:dateOfBirth} \\
& \text{rdf:type} \ rdftm:OccurrenceProperty .
\end{align*}
\]

**External occurrence (value is a URI)**

\[
\{\text{tosca, ex:synopsis, "http://www.metopera.org/synopses/tosca.html"}\} \\
\begin{align*}
_\text{a000} & \text{ ex:synopsis} \\
& \langle http://www.metopera.org/synopses/tosca.html \rangle . \\
& \text{ex:synopsis} \\
& \text{rdf:type} \ rdftm:OccurrenceProperty .
\end{align*}
\]

#### 2.6 Binary associations

A binary association corresponds closely to a single RDF statement (or a pair of statements that are `owl:inverseOf` each other), except that literals cannot participate in associations and associations have roles instead of direction.

These Guidelines define the class `rdftm:RoleProperty` and the properties `rdftm:subject-role` and `rdftm:object-role` in order to ensure correct two-way translations.

Associations with multiple signatures are not supported by these Guidelines (see Section 5.2). In addition, certain association types and properties have built-in guidance (see Section 4.3). Finally, it should be noted that while the semantics of symmetric associations (in which both role players have the same role type) are fully preserved, their translation is non-deterministic.

**Translation rules (normative)**

**TM2RDF**

- An binary association maps to a statement: the topic whose role type is the value of the association type’s `rdftm:subject-role` property becomes the subject; the association type becomes the predicate; and the topic whose role type is the value of the association type’s `rdftm:object-role` property becomes the object.

**RDF2TM**

- An RDF statement whose predicate has the properties `rdftm:subject-role` and `rdftm:object-role` becomes a binary association.
- The role of the topic that is the subject of the statement is set to the value of the `rdftm:subject-role` property, and the role of the topic that is the object of the statement is set to the value of the `rdftm:object-role` property.
role property. The role of the topic that is the object of the statement is set to the value of the rdftm:object-role property.

Examples (informative)

Binary association

```
bio:born-in( puccini : foaf:Person,
   lucca   : geo:place )

rdftm:subject-role( bio:born-in : rdftm:Relation ,
   foaf:Person : rdftm:RoleProperty )

rdftm:object-role(  bio:born-in : rdftm:Relation ,
   geo:place   : rdftm:RoleProperty )
```

Non-binary associations

Non-binary associations are mapped to an RDF pattern for describing n-ary relations described by [Noy et al] using the class rdftm:N-aryRelation and the properties rdftm:subject-role and rdftm:object-role. This pattern, which Noy et al term Introducing a new class for a relation, has two variants, both of which are supported; they differ only in whether or not the relation has a distinguished participant.

Translation rules (normative)

Check whether there now exists a vocabulary for the Noy et al patterns, and whether we can use it.

TM2RDF

- A non-binary association results in a blank node whose type is set to the association type.
- The type of this node becomes an instance of rdftm:N-aryRelation.
- A role whose type is the value of an rdftm:subject-role property results in a statement whose
subject is the role-player; the role type becomes the predicate and an instance of `rdf:tm:RoleProperty`; the blank node becomes the object.

- Each role whose type is the value of an `rdf:tm:object-role` property results in a statement whose subject is the blank node; the role type becomes the predicate and an instance of `rdf:tm:RoleProperty`; the role-player becomes the object.

**RDF2TM**

- A blank node whose type has the type `rdf:tm:N-aryRelation` becomes an n-ary association whose type is the same as the type of the blank node.
- Each statement in which the blank node is either the subject or the object and whose type is an instance of `rdf:tm:RoleProperty` results in a role, and the statement's property becomes the role type.

**Examples (informative)**

**N-ary relation with distinguished participant**

```xml
<rdf:Description rdf:about="_a000">
  < RDF:subject-role rdf:resource="#bio:killed-by"/>
  < RDF:subject-role rdf:resource="#bio:victim"/>
  < RDF:subject-role rdf:resource="#bio:perpetrator"/>
  < RDF:subject-role rdf:resource="#bio:method"/>
  < rdf:type rdf:resource="#rdftm:Relation"/>
  < rdf:type rdf:resource="#rdftm:RoleProperty"/>
</rdf:Description>
```

**N-ary relation with no distinguished participant**

```xml
<rdf:Description rdf:about="_a000">
  < RDF:object-role rdf:resource="#bio:killed-by"/>
  < RDF:object-role rdf:resource="#bio:victim"/>
  < RDF:object-role rdf:resource="#bio:perpetrator"/>
  < RDF:object-role rdf:resource="#bio:method"/>
  < rdf:type rdf:resource="#rdftm:Relation"/>
  < rdf:type rdf:resource="#rdftm:RoleProperty"/>
</rdf:Description>
```

**Unary association**

```xml
<rdf:Description rdf:about="_a000">
  < RDF:subject-role rdf:resource="#music:unfinished"/>
  < rdf:type rdf:resource="#rdftm:Relation"/>
  < rdf:type rdf:resource="#rdftm:RoleProperty"/>
</rdf:Description>
```

```xml
<rdf:Description rdf:about="_a001">
  < RDF:object-role rdf:resource="#music:work"/>
  < rdf:type rdf:resource="#rdftm:RoleProperty"/>
</rdf:Description>
```
2.8 Datatypes

The datatype features of RDF and Topic Maps are essentially compatible, except for a potential error situation if an RDF statement whose value has a datatype other than string has guidance that says it should be mapped to a topic name.

Translation rules (normative)

TM2RDF

- The datatype of a variant or an occurrence becomes the data type of a plain literal.

RDF2TM

- The datatype of a plain literal becomes the data type of either a variant or an occurrence, depending on the rules defined in Sections 3.4–3.5.

Examples (informative)

Occurrence with data type

```
{ puccini, bio:dateOfBirth, 
  [1858-12-22]^[xsd:date } 
```

```
{ a000 bio:dateOfBirt h
  '1858-12-22'^^xsd:date .
  bio:dateOfBirth 
  rdf:type rdftm:OccurrenceProperty .
```

2.9 Reification

RDF does not have a facility for reifying the relation represented by a statement, only the statement itself. These Guidelines therefore define their own pattern (based on the RDF reification vocabulary) in order to represent Topic Maps reification. The class rdftm:Relation is used for blank nodes that represent reified Topic Maps statements.

The rules defined in Sections 2.4.1 and 2.7 result in the creation of blank nodes for all variant names and non-binary associations, so no explicit reification is necessary for these constructs.

FIXME: These rules do not yet cover reified roles and topic maps.

Translation rules (normative)

TM2RDF

- A reified name, occurrence, or binary association is mapped to a statement according to the rules defined in Sections 3.4, 3.5, and 3.6, and a blank node is then created.

RDF2TM

- A blank node of type rdftm:Relation becomes a reified name, occurrence, or binary
association, unless its only additional properties are `rdftm:scope`, in which case it is treated as a scoped characteristic in accordance with the rules defined in Section 2.10.

### Examples (informative)

#### Reified name

```
[puccini *= rdfs:label : "Giacomo Puccini" =puccini-name] 
(puccini-name, dc:description, 
 [[A venerable Italian name.]])
```

```
_a000 rdfs:label "Giacomo Puccini" .
rdftm:scope rdf:type rdftm:NameProperty .

_a001
rdf:type rdftm:Relation ;
rdf:subject _a000 ;
rdf:predicate rdfs:label ;
rdf:object "Giacomo Puccini" ;
dc:description "A venerable Italian name." .
```

#### Reified occurrence

```
(puccini, bio:dateOfBirth, 
 [[1858-12-22]] =puccini-DoB 
(puccini-DoB, dc:comment, 
 [[Sometimes given as 1858-12-21.]])
```

```
_a000 bio:dateOfBirth "1858-12-22" .
bio:dateOfBirth rdf:type rdftm:OccurrenceProperty .

_a001
rdf:type rdftm:Relation ;
rdf:subject _a000 ;
rdf:predicate bio:dateOfBirth ;
rdf:object "1858-12-22" ;
dc:comment "Sometimes given as 1858-12-21." .
dc:comment rdf:type rdftm:OccurrenceProperty .
```

#### Reified binary association

```
bio:born-in( 
puccini : foaf:Person, 
lucca : geo:place ) 
~puccini-bp 
(puccini-bp, dc:comment, 
 [[The house is now a museum.]])
```

```
_a000 bio:born-in _a001 .
bio:born-in
rdftm:subject-role foaf:Person ;
rdftm:object-role geo:place .

_a002
rdf:type rdftm:Relation ;
rdf:subject _a000 ;
rdf:predicate bio:born-in ;
rdf:object _a001 ;
dc:comment "The house is now a museum." .
dc:comment rdf:type rdftm:OccurrenceProperty .
```

### 2.10 Scope

Scope provides a way to annotate a name, occurrence, or association for the specific purpose of expressing its contextual validity, and is thus a special case of making an assertion about an assertion. It is mapped to RDF using the RDFTM reification vocabulary and the property `rdftm:scope`.

#### Translation rules (normative)
TM2RDF

- A scoped name, occurrence, or binary association is reified using the rules defined in Section 2.9 and the resulting blank node is assigned one rdftm:scope property for each topic in the statement's [scope] property.
- A scoped non-binary association is translated according to the pattern described in Sections 2.7 and the node representing the reified relation is assigned one rdftm:scope property for each topic in the statement's [scope] property.
- If the scope is defined by a single topic that represents a language in the RFC 3066 vocabulary, the rules for translating language tags (given in Section 2.10.1) should be applied instead.

RDF2TM

- A blank node of type rdftm:Relation (or its subtype rdftm:N-aryRelation) with one or more rdftm:scope properties becomes a scoped statement.

Examples (informative)

Scoped name

```
[ puccini = foaf:name : "Giacomo Puccini"
 / ex:foo ex:bar]
```

 Scoped occurrence

```
{tosca, music:synopsis,
 "http://www.metopera.org/synopses/tosca.html"}
 / opera:metopera
```

 Scoped association

```
music:infl-by( butterfly : ex:object, iris : ex:agent )
 / carner
rdftm:subject-role{
 music:infl-by : rdftm:Relation ,
 ex:object : rdftm:RoleProperty }
rdftm:object-role{
 music:infl-by : rdftm:Relation ,
 ex:agent : rdftm:RoleProperty }
```

 Scoped name with variant

```
[boito = foaf:name : "Tobia Gorrio"
 / ex:nom-de-plume
 ["gorrio, tobia" / tm:sort]]
```

2.10.1 Language tags

RDF language tags are semantically equivalent to names or occurrences that are scoped by a natural language using codes defined by [RFC 3066]. These Guidelines define a namespace for such language identifiers to be used when translating between RDF and Topic Maps.

Translation rules (normative)

TM2RDF

- A name or internal occurrence that is scoped by a single topic whose subject identifier is in the RDFTM Language namespace translates to an RDF statement with a corresponding language tag, where the tag is the name portion of the language topic's subject identifier.

RDF2TM

- An RDF plain literal with a language tag becomes the value of a name or occurrence according to the rules defined in Sections 3.4–3.6, which is scoped by a single topic representing the corresponding language. The subject identifier of the language topic is formed by appending the lower-cased form of the language tag to the namespace URI http://www.w3.org/2006/rdftm/lang/.

Examples (informative)

Name scoped by language

```
#PREFIX lang
@"http://www.w3.org/2006/rdftm/lang/"
{ [la-fanciulla
   = dc:title : "La Fanciulla del West"
   = dc:title : "The Girl of the Golden West"
   / lang:en-us] } / lang:en-us
```

Occurrence scoped by language

```
#PREFIX lang
@"http://www.w3.org/2006/rdftm/lang/"
{ [la-fanciulla,
   dc:description,
   "En opera av Puccini hvor handlingen
   finner sted i Amerika."]
} / lang:nor
```
3. Authoring guidelines

The following guidelines are offered to authors of topic maps and RDF documents who wish to ensure maximum interoperability.

3.1 Guidelines for authors of topic maps

1. Always assign a subject identifier to typing topics.
2. Ensure that association types have `rdftm:subject-role` and `rdftm:object-role` guidance statements.
3. Do not use association types with multiple signatures.
4. Ensure that typing topics are not used to type more than one kind of statement (i.e., name, occurrence, or association).

3.2 Guidelines for authors of RDF vocabularies

1. Ensure that every property in the vocabulary is either declared to be an `rdftm:NameProperty` or `rdftm:OccurrenceProperty`, or else has `rdftm:subject-role` and `rdftm:object-role` properties.
2. Ensure that the datatype of every `rdftm:NameProperty` is `xsd:string`.

4. The RDFTM vocabulary

To be rewritten as OWL ontology.

The `rdftm` vocabulary consists of the following classes and properties:

4.1 RDFTM classes

**InformationResource**

The class of resources that can be represented as a sequence of bytes and thus could potentially be retrieved over a network. A subclass of `rdf:Resource`.

**N-aryRelation**

The class of n-ary relations; a subtype of Relation.

**NameProperty**

The class of properties that have the semantics of assigning a name.

**OccurrenceProperty**

The class of properties that have the semantics of assigning an occurrence.

**Relation**

The class of relations.

**RoleProperty**

The class of properties that have the semantics of describing roles in associations.

**Variant**

[...]

11/12/2007
4.2 RDFTM properties

**itemIdentifier**

[...]

**objectRole**

A property that specifies the type of role to be played by the object of an RDF statement that is to be translated to an association; or (optionally) the type of the role player that becomes the object of the RDF statement when translating from an association.

**scope**

A property that specifies (part of) the context in which a statement is considered to be valid.

**subjectRole**

A property that specifies the type of role to be played by the subject of an RDF statement that is to be translated to an association; or the type of the role player that becomes the subject of the RDF statement when translating from an association.

**subjectIdentifier**

A property that assigns a subject identifier to a topic or resource.

**value**

[...]

**variant**

A property that specifies an alternative form of a name.

4.3 Built-in guidance

Rewrite as table with rules expressed in both TM and RDF.

The following guidance rules are built-in:

1. `tm:type-instance`
   - `rdftm:subjectIdentifier rdfs:type ;`
   - `rdftm:subject-role tm:instance ;`
   - `rdftm:object-role tm:type .`

2. `tm:supertype-subtype`
   - `rdftm:subjectIdentifier rdfs:subClassOf ;`
   - `rdftm:subject-role tm:subtype ;`
   - `rdftm:object-role tm:supertype .`

3. `rdfs:label`
   - `rdf:type rdftm:NameProperty .`

4. `rdftm:subject-role`
   - `rdftm:subject-role rdftm:Relation ;`
   - `rdftm:object-role rdftm:RoleProperty .`
   - `rdftm:object-role`
4.4 The RDFTM Language vocabulary

This document defines a vocabulary with the namespace http://www.w3.org/2006/rdftm/lang/ for languages covered by [RFC 3066]. URIrefs in the vocabulary are used as subject identifiers for language topics. They are formed by appending the lower-case form of the RFC 3066 code to the URI of the namespace. Section 2.10.1 describes how the subject identifiers are used; the following table shows examples of RDF language tags and their corresponding subject identifiers:

<table>
<thead>
<tr>
<th>Language</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lang:en</td>
<td>en</td>
<td>Single, two-letter subtag based on ISO 639</td>
</tr>
<tr>
<td>lang:en-us</td>
<td>en-US</td>
<td>2-part tag based on ISO 639 and ISO 3166</td>
</tr>
<tr>
<td>lang:en-us</td>
<td>en-us</td>
<td>Equivalent 2-part tag using different casing</td>
</tr>
<tr>
<td>lang:nor</td>
<td>nor</td>
<td>Single, three-letter subtag based on ISO 639-2</td>
</tr>
<tr>
<td>lang:sgn-us-ma</td>
<td>sgn-US-MA</td>
<td>3-part tag for Martha’s Vineyard Sign Language</td>
</tr>
</tbody>
</table>

5. Limitations

This section describes the limitations of these Guidelines in terms of non-deterministic results and unsupported constructs.

5.1 Non-deterministic rules

- Symmetric associations in TM2RDF — because there is no deterministic way to choose the subject and object of the resulting statement.

5.2 Unsupported constructs

- Typing topics that have no subject identifier — because they cannot be translated to a property with a URIref.
- Association types with multiple signatures — because there is no simple way to express the rules for assigning subject and object roles.
- Reified association roles — because there is no construct in RDF that is equivalent to an association role.
- Reified topic maps — because there is no construct in RDF that is equivalent to a topic map.
- Topic maps in which the same type is used to type different kinds of statements — because it would lead to ambiguity when roundtripping.
- In RDF2TM, plain literals with datatypes other than xsd:string that are the value of properties of type rdftm:NameProperty — because topic names must be strings.
6. Acknowledgements

[TBD]

7. References

[Garshol]  
Garshol, Lars Marius: *Living with Topic Maps and RDF*,  

[Gentilucci et al]  
Gentilucci, Riccardo; Pirruccio, Marco: *Metainformazioni sul World Wide Web: Conversione di formato e navigazione*, University of Bologna, Masters Thesis, (2002; in print; in Italian)

[LTM]  
Garshol, Lars Marius: *The Linear Topic Map Notation: Definition and introduction, version 1.2*,  

[Noy et al]  

[OWL-ref]  

[OWL-sem]  

[Pepper]  
Pepper, Steve: *The TAO of Topic Maps: Finding the Way in the Age of Infoglut*,  

[RDF Primer]  

[RDF Schema]  

[RDF Semantics]  

[RFC 3066]  
Alvestrand, H. (ed.): *RFC 3066: Tags for the Identification of Languages*,  
[Survey]

[TMDM]

[Turtle]