RDF and TopicMaps
An Exercise in Convergence

Graham Moore
Vice President Research & Development, Empolis GmbH
gdm@empolis.co.uk
http://www.empolis.co.uk
http://www.empolis.com

XML Europe 2001 Berlin

empolis
BERTELSMANN MOHN MEDIA GROUP

Introduction

There has long been a sense in the semantic web community that there is a synergy between the work of ISO and TopicMaps.org on TopicMaps and that of the W3C on RDF. This paper looks at why and how we can bring these models together to provide a harmonised platform on which to build the semantic web.

The reasoning behind bringing together these two standards is in the fact that both models are intent on describing relationships between entities with identity. The question we look to answer in this paper is 'Is the nature of the relationships and the identified entities the same'. If we can show this to be true then we will be able to have a common model that can be accessed as a TopicMap or as a RDF Model. To make this clearer, if we have a knowledge tool X we would expect to be able to import some XTM syntax, some RDF syntax and then run either a RDF or TMQL query in the space of tool X and expect sensible results back across the harmonised model.

In order to achieve this aim we need to show a model to model mapping between the two standards. We present the TopicMap model, the RDF model, a discussion on modelling versus mappings and then a proposed mapping between the two. As part of the mapping we make suggestions as to the changes that could be made to better support interoperation and finally we conclude and provide an insight into future work.

The TopicMap Model

Before we can discuss the ways in which we can map the RDF and TopicMap model we must first present our understanding of these models. The model is the internal representation of a TopicMap or RDF model after parsing syntax or using an API to create them. This document does not discuss how a semantically consistent model is defined, see XTM Processing Model and RDF specifications for more details on this. Here we explicitly define the data models that capture the isness of the two paradigms. The TopicMap model is presented as a set of classes with properties and relationships. The diagram below expresses the core TopicMap model.

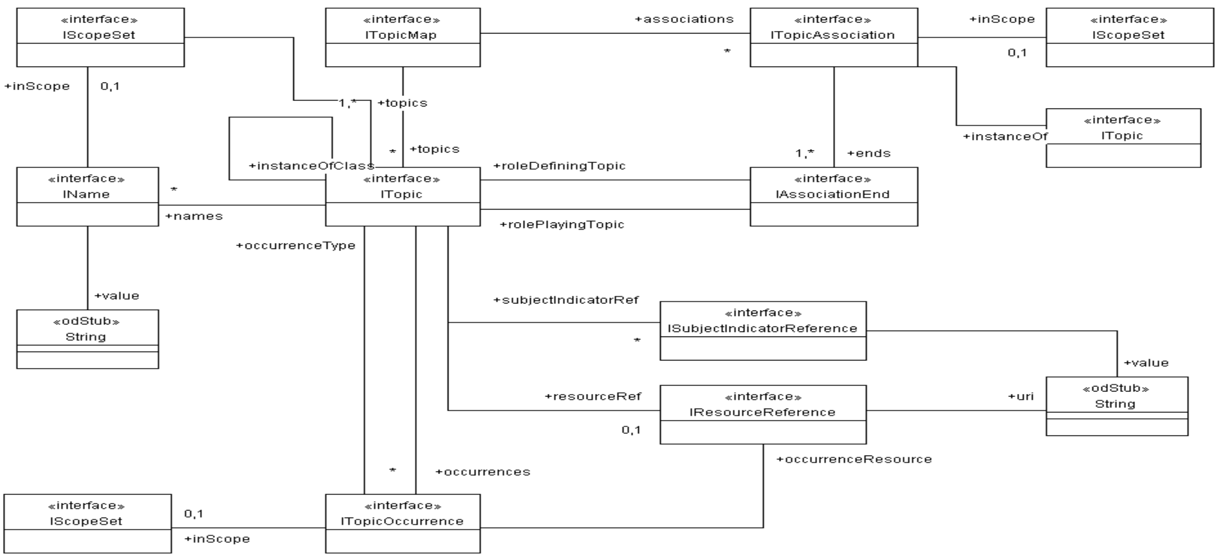

Figure 1. The Core Topic Map Model.

The core data model shows topics with identity, occurrences and names. It also shows topic associations and association ends (also called 'members'). ScopeSets are used to constrain the context of names, associations and occurrences. A more detailed description of each entity follows.

TopicMap
The TopicMap is the organising principal of the model. A single TopicMap instance will manage many topics and topic associations. The TopicMap has two properties, *topics* and *associations*.

Topic
There are only a few concepts in the model but at the heart of all of them is the topic. The topic is the proxy for some concept, idea, whatever, some thing. As defined in XTM, the topic has names, occurrences, identity and plays roles in associations. All of these aspects are represented in the model.

The topic has the following properties:

- *names* - the name constructs that help form the identity of a topic.
- *instanceOfClass* - the set of topics that are classes of which this topic is an instance.
- *occurrences* - the occurrence constructs that reference resources related to this topic.
- *SubjectIndicatorRefs* - the unique identity constructs that help to define the identity of this topic.
- *ResourceRefs* - if this topic is itself a proxy for a resource then its identity consists of the resource reference that identifies the resource.
- *AssociationEnds* - the set of association ends that bind this topic into an association.

TopicAssociation
The topic association is the entity that binds the map together in conjunction with the *AssociationEnd* entity. The topic association has the following properties:

- *Ends* - the set of association ends that bind topic into this association.
- *inScope* - the *ScopeSet* that constrains the context in which this association holds.
- *instanceOf* - the *Topic* entity that is the class, of which this association is an instance.

AssociationEnd
The association end is contained by an association and links together a role defining topic and a role playing topic. The association end can be accessed from both associations and the topic that is the role playing topic. The *AssociationEnd* has the following properties:

- *Association* - the association that this end is a member of.
- *roleDefiningTopic* - the topic that defines the role in the association.
- *rolePlayingTopic* - the topic that plays the role in the association.

ScopeSet
Several aspects of the model rely on the concept of scope to define a context in which they are valid. Names and Associations are examples of this. In the model, scope is represented by an aggregated entity the ScopeSet. The ScopeSet consists of the set of topics that together are the scope. A ScopeSet has just one property in the core data model and that is *topics* which returns the set of topics that comprise it.

TopicOccurrence
The topic occurrence is the structure that associates resources with topics. The properties of the occurrence are :

- *topic* - the topic that this occurrence applies to.
- *occurrenceInstanceOf* - the topic that defines the class of which this occurrence is an instance.
- *resourceReference* - the resource reference that can be used to locate the occurrence resource.
- *inScope* - the scope that defines the context in which the occurrence is valid.

SubjectIdentityReference
The subjectidentity entity has just a single property value that is a string that helps to identify the topic to which the subject indicator reference belongs.

ResourceReference
The resource reference entity has just a single property URI that is a string that locates a resource.

There is a lot of commonality even in this small model providing scope for further rationalisation. E.g. occurrences and names are refinements of topic associations. And as associations are the most commonly reified construct that topic association derives from topic. This possible hierarchy is presented below as a point of information.

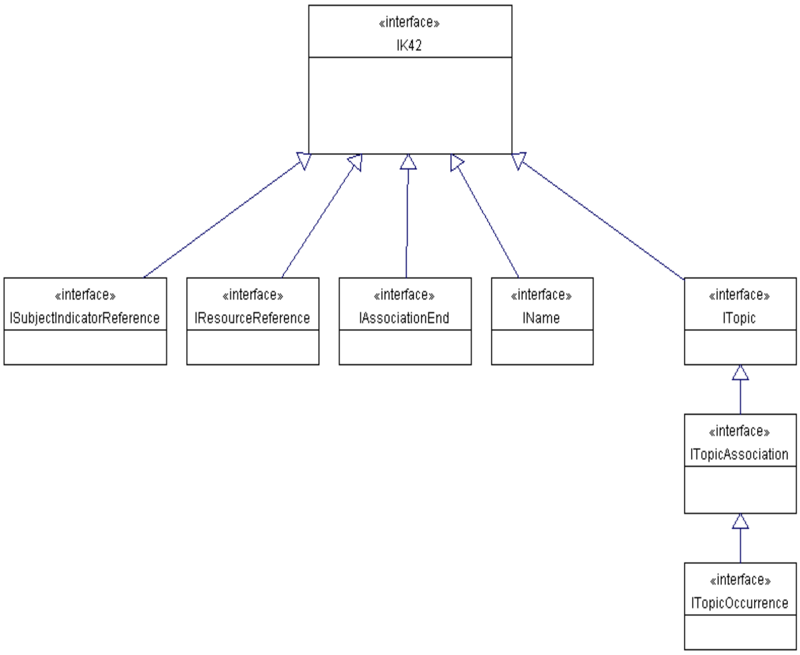

Figure 2. TopicMap - a hierarchy of entities.

This hierarchy has an additional feature at the top of the model that is a general TopicMap entity. Due to reification it is necessary to have a common entity which can be reified. Of course the reifying property can be flattened out for each TopicMap entity, but this illustrates some of the commonality of the model.

The next figure illustrates the ability to reify TopicMap constructs.

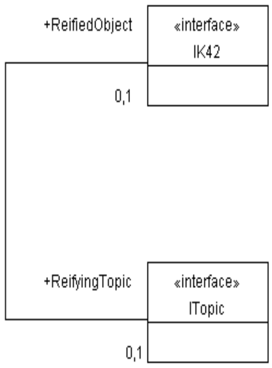

Figure 3. The reification construct.

This section has presented the core TopicMap model as a set of entities and relationships. We have not presented here how these models are constructed from the XTM or ISO syntax but have instead defined a data model that can be used for the internal representation of TopicMaps. Work not presented here is the advanced view of occurrences and names as topic associations and the mapping to the syntax.

Having this model will enable us to define rigorous work. Examples of this are TMQL and other TopicMap related efforts. In addition, and relative to this paper, we can use this model in our efforts of model convergence.

The RDF Model

Above we described the TopicMap model, before we can begin a presentation of the work on convergence we first need to expose the core RDF model.

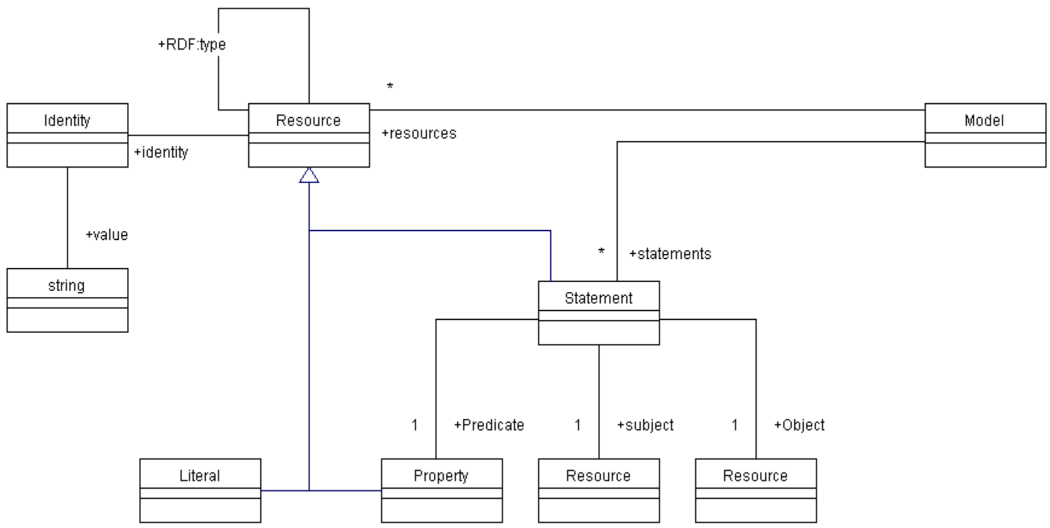

Figure 4. The RDF Model

The figure above shows the RDF model as a set of entities and relationships. Note that we have not added in the any aspects of RDF Schema as these constructs are defined in terms of this basic model. Thus if we can find equivalents and mappings at this level then we implement the semantics of RDF Schema on top. Below is a more detailed explanation of the entities and their relationships.

Model
The Model is the binding point for a collection of Statements. Given that statements are comprised of resources, it is not a surprise that the model also has access to all the resources utilised within the system. The Model has two properties, *statements* and *resources* returning the set of statements and resources respectively.

Resource
The resource is at the hub of the RDF data model. It is used in the main data structure *statement* and is also used self-referentially to define type. The resource has two key properties, *identity* and *type*. The *identity* property returns an object that describes the identity of a resource, the *type* property returns a resource of type class, of which this resource is an instance.

Identity
The identity entity is used to define the uniqueness of a given resource. All resources have some form of identity within the system. This entity has a single property, *value*, that returns the string value that defines the identity.

Property
The property is a refinement on the general resource. Only property resources can be used as the predicate in a statement. Property has no properties other than those defined on *resource*.

Statement
The statement is a key binding principle in the RDF data model, it defines a triple of a predicate (the property), a subject and an object. It can be read that the property is a property of the subject and that the value of that property is the object. In OO terms, it is the subject that is the object which has a property where the value of that property is another object. The statement has three properties in addition to those specified on Resource. The properties are *Predicate,* S*ubject* and *Object*. *Subject* and *Object* return the resources that play those roles, the *Predicate* returns a property that defines the arc between the Subject and Object.

Literal
Literals are here for completeness. It is possible to use a Literal as a value in a statement in the Object role.

The last two sections have described the TopicMap and RDF models in such as way that we can now describe the relationships and mappings that exist between these two models. The first investigation looks at how the two models can be mapped and interchanged in a way we call modelling the model. Modelling the model is where we define a model using the constructs of another modelling language.

Mapping the Models Vs Modelling the Models

In the introduction we define a clear goal that we should be able to run a TMQL query against an RDF model and get 'expected results' i.e. those that would be gained from running a query against the equivalent TopicMap. To make this possible we need to make the models map rather than using the models to describe each other.

The key difference in these approaches is that one provides a mapping that is semantic, the other uses each standard as a tool for describing other models. It is interesting that both models are flexible enough and general enough to allow each to be modelled using the other. Below is a few example of the kind of constructs that can be modelled. We show RDF statements being modelled as topic associations, and Topic Associations being modelled as set of RDF statements.

RDF modelled using TopicMap Constructs

We first need to define some 'semantic' topics. These are the topics that will allow us to interpret the TopicMap as a RDF model. For topics we use a refined syntax:

```
<topicname : [PSI]>
```

We use the TopicMaps PSI mechanism to uniquely identify the topic.

RDF Topics:

```
(resource  : 'http://www.empolis.com/rdftmmapping#rdf-resource')
(statement : 'http://www.empolis.com/rdftmmapping#rdf-statement')
(property  : 'http://www.empolis.com/rdftmmapping#rdf-property')
(subject   : 'http://www.empolis.com/rdftmmapping#rdf-subject')
(object    : 'http://www.empolis.com/rdftmmapping#rdf-subject')
(identity  : 'http://www.empolis.com/rdftmmapping#rdf-resource')
(literal   : 'http://www.empolis.com/rdftmmapping#rdf-literal')
(model     : 'http://www.empolis.com/rdftmmapping#rdf-model')
```

With this set of topics we are able to construct topic associations that convey a RDF model. Again we used a refined notation to succinctly convey the intent.

The general form for the associations is the PSI for the association instance topic and then many tuples of role defining topic and role playing topic.

```
< [topicassocinstanceofPSI ], <[RDTPSI], [RPTPSI]>,  … >
```

Where the most common RDF statement is modelled as a Topic Association in the form:

```
< 'http://www.empolis.com/rdftmmapping#rdf-statement' ,
       'http://www.empolis.com/rdftmmapping#rdf-subject', ROLEPLAYER
       'http://www.empolis.com/rdftmmapping#rdf-property', ROLEPLAYER
       'http://www.empolis.com/rdftmmapping#rdf-subject', ROLEPLAYER >
```

and ROLEPLAYER in each case in as yet undefined.

So to express the RDF triple

```
       ('graham moore', 'works-for', 'empolis')
```

We require that all resources are identified, even literals. We are aware of issues with this interpretation of literals, thus we define it as:

```
< 'http://www.empolis.com/rdftmmapping#rdf-statement' ,
       'http://www.empolis.com/rdftmmapping#rdf-subject', PSIGRAHAMMOORE
       'http://www.empolis.com/rdftmmapping#rdf-property', PSIWORKSFOR
       'http://www.empolis.com/rdftmmapping#rdf-subject', PSIEMPOLIS >
```

As we have the 'statement' topic we can now ask the topic map to give us a list of all statements - just the same as we could a RDF model. Likewise we could retrieve a 'subject' view of all the properties of a given resource.

TopicMaps modelled using RDF

In this section we show how the TopicMap model can be expressed using RDF constructs. For syntactic conciseness we define our RDF triples as follows:

```
< subject, property, object >
```

In the same way that we defined some 'semantic' topics for the RDF model in TopicMaps we also need to define some URI RDF resources that we understand to be TopicMap resources.

```
'http://www.empolis.com/rdftmmapping#tm-topic'
'http://www.empolis.com/rdftmmapping#tm-topicassoc'
'http://www.empolis.com/rdftmmapping#tm-instanceof'
'http://www.empolis.com/rdftmmapping#tm-topicassocmember'
'http://www.empolis.com/rdftmmapping#tm-roleplayingtopic'
'http://www.empolis.com/rdftmmapping#tm-roledefiningtopic'
'http://www.empolis.com/rdftmmapping#tm-topicoccur'
'http://www.empolis.com/rdftmmapping#tm-topicname'
'http://www.empolis.com/rdftmmapping#tm-topicnamevalue'
'http://www.empolis.com/rdftmmapping#tm-scopeset'
'http://www.empolis.com/rdftmmapping#tm-subjindicatorref'
'http://www.empolis.com/rdftmmapping#tm-resourceref'
```

Using the above resources we can create the TopicMap model using RDF.

So to model the topics:

```
('graham moore', 'http://www.empolis.com/people/gdm')
('empolis', 'http://www.empolis.com/company/empolis')
('employment', 'http://www.empolis.com/asssocs/employment')
('employer', 'http://www.empolis.com/roletypes/employer')
('employee', 'http://www.empolis.com/roletypes/employee')
```

and the association "the topic 'graham moore' plays the role of 'employee' in the association 'employment' where 'empolis' plays the role of 'employer'" we define as the RDF triples:

```
( http://www.empolis.com/people/gdm
  http://www.empolis.com/rdftmmapping#tm-topicname,
  resource:id:x23)

( resource:idx23,
  http://www.empolis.com/rdftmmapping#tm-topicnamevalue
  'Graham Moore')

# We need the anonymous 'resource:id:x23' as the topicname construct is not
atomic and we need to make statements about it. Note also that the identity/
URI of the resource that is the subject is the same as the subject indicator
reference of the topic. This is not necessary, in the same way as the
topicname is defined as resource with a generated identity the topic 'Graham
Moore' can have the same identity.

( http://www.empolis.com/people/gdm
  http://www.empolis.com/rdftmmapping#tm-subjindicatorref',
  'http://www.empolis.com/people/gdm')
```

```
# We define the other topics is a similar fashion and then the association
is defined as follows:

( resource:id:assoc:1,
  http://www.empolis.com/rdftmmapping#tm-instanceof,
  'http://www.empolis.com/asssocs/employment')

( resource:id:assoc:1,
  'http://www.empolis.com/rdftmmapping#tm-topicassocmember'
  resource:id:assocmember:1)

( resource:id:assoc:1,
  'http://www.empolis.com/rdftmmapping#tm-topicassocmember'
  resource:id:assocmember:2)

( resource:id:assocmember:1,
  http://www.empolis.com/rdftmmapping#tm-roledefiningtopic,
  http://www.empolis.com/roletypes/employee)

( resource:id:assocmember:1,
  http://www.empolis.com/rdftmmapping#tm-roleplayingtopic,
  http://www.empolis.com/people/gdm)

( resource:id:assocmember:2,
  http://www.empolis.com/rdftmmapping#tm-roledefiningtopic,
  http://www.empolis.com/roletypes/employer)

( resource:id:assocmember:2,
  http://www.empolis.com/rdftmmapping#tm-roleplayingtopic,
  http://www.empolis.com/company/empolis)
```

Above we demonstrated how we can capture the key aspects of both models in terms of the constructs of the other. The key aspects of note are that a Topic was used to define the arc, and an association with three members constituted a statement in the TM RDF model and that multiple RDF statements could be used to recreate the TopicMap model.

This approach is useful in a sense as it provides a mechanism for moving from one domain into another. But it assumes that we are always working in one domain or the other. With this approach we can run an RDF query against an RDF model where that model has been defined using topic map constructs, but that doesn't make it a TopicMap.

It is an interesting property of levels and semantic interpretation that means that a set of RDF triples can be interpreted as a TopicMap and likewise that an RDF statement can be seen in a collection of topics and associations. It is conceivable that we could continue the cycle further, having TopicMaps modelled using RDF where the RDF is in fact defined as topics and associations. This ability to place semantics and meaning onto 'identified' entities is why both these models are so powerful in letting us express knowledge and relationships and it is this that drives us towards finding a common model at the heart of the two.

In this section we have shown how TopicMaps can be used for modelling RDF and vice versa, this is an interesting result on its own but doesn't help us achieve our aim of general interoperability of the TopicMap and RDF models. The next sections illustrate the essence of the TopicMap and RDF models. We now go on to present a mapping between the two.

A Model to Model Mapping

The last section presented how the two models can be used to model each other. We concluded that this was a valuable result in that it showed the flexibility of the two models and it also provided a way to define RDF that could be interpreted as a TopicMap and vice versa. What this approach did not show was how the two models directly corresponded to one another. We propose that this kind of mapping would enable RDF that had not been created with special, known resources - as we did above, to be processed and understood as a TopicMap. Similarly, that a TopicMap could be processed and then queried as a set of RDF triples, again without the aid of 'semantic' topics or resources.

This section presents an approach for mapping the two models onto one another in a direct form. Before we present the mapping we should make an observation that has helped guide us in trying to understand the relationship between TopicMaps and RDF. RDF is concerned with describing the ***arcs between*** entities with ***identity***, TopicMaps is concerned with describing ***typed relationships*** between entities with ***identity***.

The RDF & TopicMap Model Mapping

The RDF model has less constructs than the TopicMap model so we are going to approach the mapping from the point of view of the RDF model and define the TopicMap equivalents. In addition we will show how the apparently additional machinery of TopicMaps can be viewed in the RDF model.

RDF Model
The RDF model entity plays a similar role to the TopicMap, it is the organising principle of the model. In this mapping we will define equivalence between the two.

Identity
The RDF identity maps cleanly onto the SubjectIndicatorReference in the TopicMap model. We should perhaps comment here that TopicMaps distinguishes between identities of topics that 'are' resolvable resources and those which represent non addressable subjects. In terms of pure identity it is unnecessary at this stage to make the distinction.

Resource
Given identity maps onto the SubjectIndicatorReference it seems appropriate that the RDF resource maps onto the TopicMap Topic entity. While there may be some debate about the nature of resource it seems clear that the intent of these two 'entities' is to be able to talk about 'things'. The identity aspect is the main reason for us to assume this fit. Identity is about creating a space in which you can talk about a thing.

RDF Statement
The most challenging aspect of creating a consistent model between RDF and TopicMaps is in the area of associations. What we mean by this is what does a RDF statement look like as a Topic Association and conversely what does a topic association look like in terms of an RDF statement or statements. The choices we have made here are based on the properties identified earlier, that topic associations are the description of topics playing roles in typed associations whereas RDF statements are the description of an arc from one entity to another.

We had the feeling that there is no 'complete' mapping of the models but there is enough to make it a worthwhile endeavour. We feel this because of simple observations that RDF has three pieces of information and TopicMap associations have five. The RDF model has two entities and an arc, and in a binary association a Topic Association is comprised of the association type, two role defining topics and two role playing topics. Something has to give.

The rest of this paper is presented in two parts, firstly, representing RDF statements as Topic Associations and secondly considering Topic Associations as RDF statements. The first of these sections concludes with a small set of recommended changes to TopicMaps to better enable clean semantic interoperation with RDF and vice versa. It is interesting that the changes need only to be in one model to enable better two way mapping.

RDF Statements as Topic Associations

We will consider that the RDF statement is equivalent to an anonymous topic association, the subject and object resources are in fact role playing topics in the anonymous association, the role defining topics are in fact either defined as 'Topic' or perhaps 'Subject' and 'Object', and the 'arc' between the topics is defined by the RDF statement property.

It has already been stated though that arcs are not part of the topic map model. No they aren't, but there are TopicMap mechanisms to cater for them. A simple example uses scoped names to define the arcs between role playing topics in a topic association.

The next diagram shows an example of how a RDF statement can be viewed as a Topic Association.

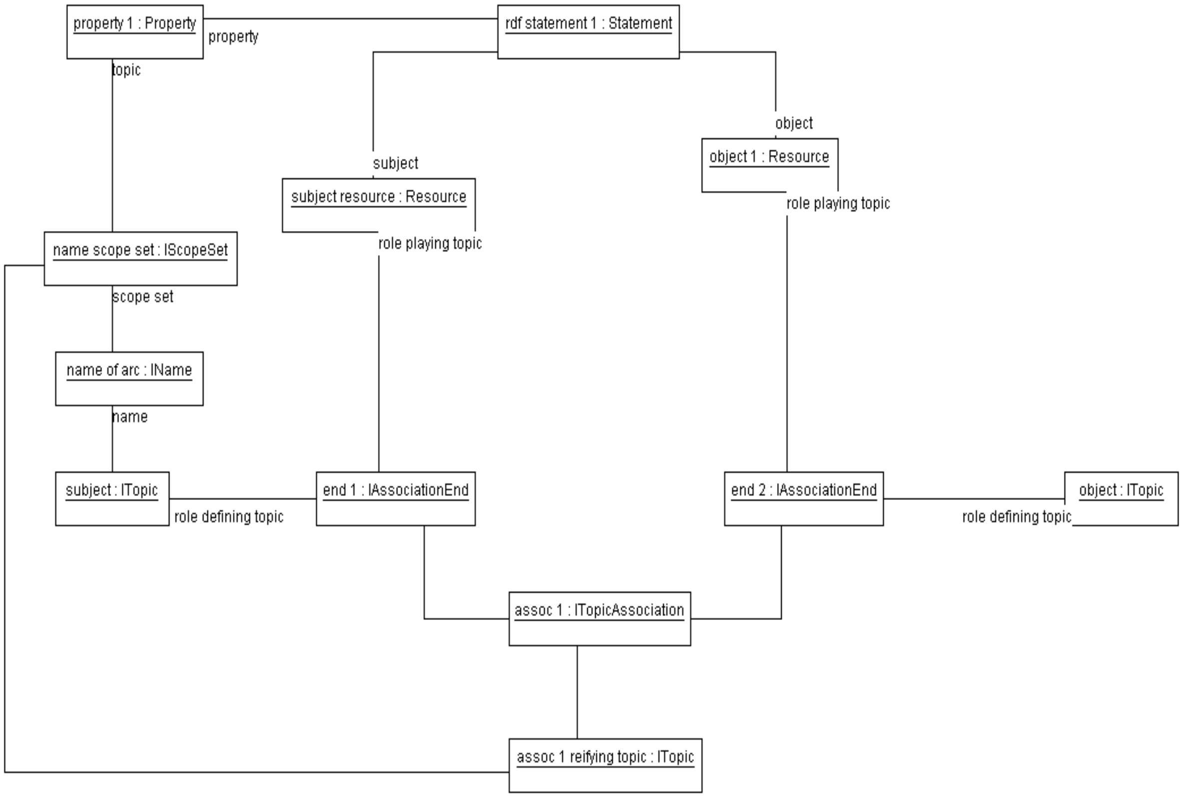

Figure 5. An RDF statement viewed as a TopicAssociation.

In the mapping above we consider that the topic association is equivalent to the RDF statement and that the Subject Resource and Object Resource play the 'subject' and 'object' roles within the association. We then use a mechanism on the name of the role defining topic, that enables us to define the 'arc' between the two topics playing roles in this assoc. This TopicMap name is the value of the property in the RDF triple. To preserve the identity of the property from the RDF it is also possible that the name is reified by a Topic that has the identity of the Property.

The problem with this approach is that the 'arc' is not a first class entity in the TopicMap model. It is proposed here that TopicMaps adopt the concept of 'arc' as a first class part of the model to bring it in line with XLink and RDF. This would mean that there was a TopicMap construct with names and identity that described the direct arc between two topics within a particular association.

There is an additional value that creating the 'arc' as a concept will have. Once there are templates in TopicMaps it will be possible to define the role defining topics, the classes of role playing topic and the arcs between the roles. These arcs will have identity. Thus an RDF statement that had a particular 'property' where that property has identity would be able to map into a TopicMap template. Given this, it would be possible to instantiate the role defining topics as being something more 'semantic' than just 'Object' and 'Subject'.

Thus currently, RDF statements can be seen as Topic Associations but the extent to which we can make useful queries against them is still limited. The addition of 'Arcs' and 'Topic Association Template' will enable a full and powerful integration of these standards.

Below is the extended TopicMap Model that supports both Topic Association Templates and Arcs.

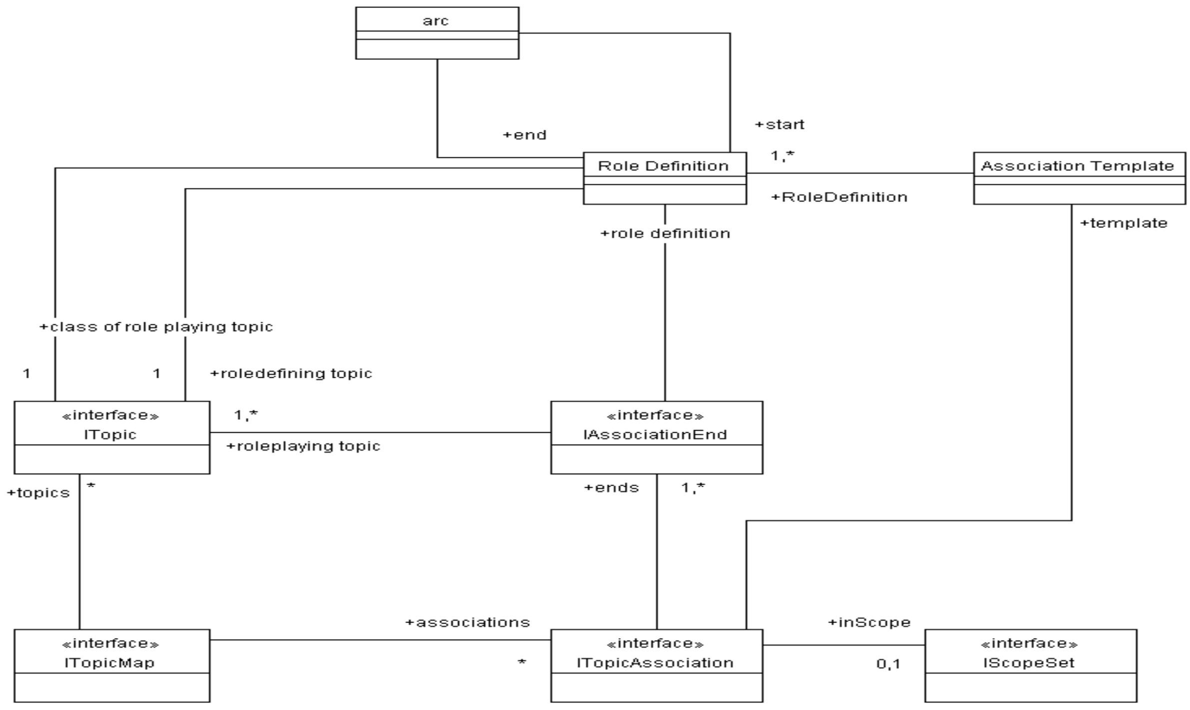

Figure 6. Extended Topic Map Model to Support Templates and Arcs

It is proposed that by extending the Topic Map model as outlined above the ability to move between semantically valid RDF and TopicMaps will be enabled.

Topic Associations as RDF

In this section we present Topic Associations as RDF statements. We adopt a similar approach to viewing associations as that taken in XLink i.e. an RDF statement exists for each 'arc' between topics in the association. Thus, for a binary Topic Association there are two RDF statements.

The following diagram shows how a binary topic association can be viewed as two RDF statements. In the diagram only one of the statements has been completed. However, there is a symmetry that should enable the reader to fill in the missing connections and entity instances.

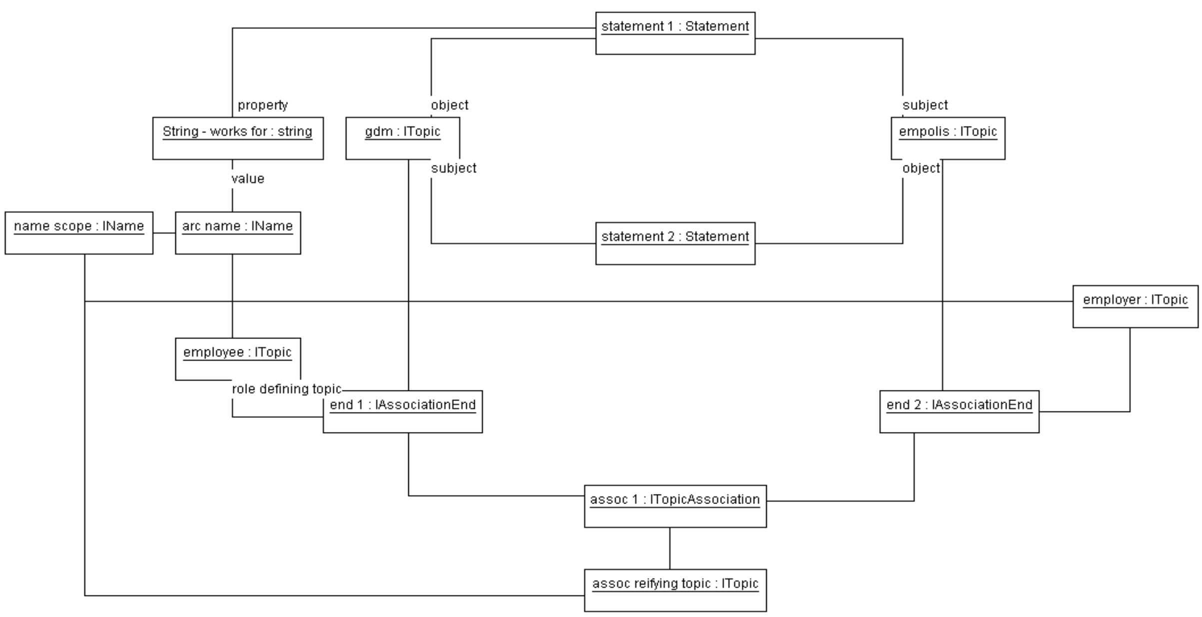

Figure 7. A Topic Association as RDF Triples.

The above figure shows how the RDF triples are composed of each topic being both the subject and object in two separate RDF triples. Again, the enhancement described above will provide a more useful connection of the semantic of association - in that the arcs will be meaningful in both TopicMaps and RDF.

This section has shown how the main constructs of the RDF model can be expressed in TopicMaps and vice versa. It has tackled the key issues of the mapping which lie around the area of Associations and Statements.

Conclusions and Future Work

This paper has set out to understand how the RDF and TopicMap models can inter-operate at a fundamental level. We stated that we wanted to be able to view TopicMaps as RDF models and RDF models as TopicMaps. We presented both the TopicMap and RDF models and then described the way in which each language can be used to model the other. This is an interesting result that shows the generality of both of the models and proved that both standards are concerned with defining relationships between entities with identity.

We then went on to investigate how the two models could be mapped into one another. While we found there was a useful mapping that could be performed it was felt that some additions to the TopicMap model - Templates and Arcs would enable two way transition from RDF to TopicMaps and vice versa.

We conclude that making some non-regressive enhancements to TopicMaps would enable a useful degree of convergence between TopicMaps and RDF, creating a single common semantic space in which to define the semantic web.