

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221211437>

# Building Topic Maps in OWL-DL.

Conference Paper · January 2005

Source: DBLP

---

CITATIONS

12

---

READS

130

1 author:



Anne Cregan

17 PUBLICATIONS 276 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Symbol Grounding [View project](#)



Overview of Semantic Technologies [View project](#)

## Building Topic Maps in OWL-DL

---

Anne Cregan  
*University of New South Wales*

---

### **Abstract**

Both ISO's Topic Map Standards and the W3C's Semantic Web Recommendations provide the means to construct meta-level semantic maps describing relationships between information resources. Developed independently, attempts at interoperability between the original Topic Map standard and RDF have proved challenging. However, ISO 13250's drafting of an explicit Topic Map Data Model (TMDM) early in 2005, combined with the advent of the W3C's more expressive Web Ontology Language (OWL) Recommendations in 2004, together now provide the possibility for authoring TMDM-conforming Topic Maps directly in OWL. OWL provides the ability to express the TMDM constraints explicitly and to ensure that Topic Maps authored in OWL conform to the TMDM.

This paper presents a construction of the TMDM model as an OWL-DL ontology. This "TMDM Ontology" is a construction in OWL of the Topic Map concepts modelled by the TMDM, making them available for use by Topic Map authors as a basis for building TMDM-compliant Topic Maps directly in OWL. Using OWL-DL as the language for Topic Map authoring gives users access to OWL's formal semantics, constraint expressivity, and suite of tools such as Protégé, which includes an API and capability for ontology visualisation, querying, and automated constraint checking and reasoning using Description Logic reasoners. The approach described does not require the use of any Topic Map language, or Topic Map Constraint or Query language, although a simple algorithm could translate the OWL-authored Topic Maps directly into other TMDM-based Topic Map authoring languages if required, providing access to Topic Map engines and tools.

Illustrating by example, a Topic Map written in OWL-DL using Protégé is shown, highlighting the constraint and querying abilities provided, which overlap many of the requirements set by the ISO for a Topic Map Constraint Language and Query Language. One outstanding issue regarding the interpretation of Typing is identified, and options for its resolution are discussed.

# Building Topic Maps in OWL-DL

## Table of Contents

1 Motivation.....	1
2 Scope of the Approach.....	1
3 Background.....	1
3.1 The ISO's Topic Map Tradition.....	1
3.1.1 Topic Map Origins.....	1
3.1.2 Incomplete Specification of Topic Maps.....	2
3.1.3 A Formal Data Model for Topic Maps.....	2
3.1.4 The TMDM is not the last word on Topic Maps .....	2
3.2 The Semantic Web Initiative.....	2
3.2.1 Berners-Lee's Vision.....	2
3.2.2 Formal Semantics.....	2
4 Interoperability Issues.....	2
4.1 Topic Maps and RDF.....	2
4.1.1 Interoperability Challenges.....	2
4.2 TMDM and OWL.....	3
4.2.1 New Interoperability Possibilities.....	3
4.2.2 TMDM to OWL? .....	3
4.2.3 OWL to TMDM?.....	3
5 TMDM to OWL Mapping Investigation .....	3
5.1 The TMDM side.....	3
5.1.1 What does the TMDM give us? .....	3
5.1.2 But wait, lurking in the back of the TMDM closet.....	4
5.2 The OWL side .....	5
5.2.1 Flavours of OWL.....	5
5.2.2 Which flavour of OWL tastes best to make Topic Maps?.....	5
5.2.3 What can be asserted in OWL-DL?.....	6
5.2.4 OWL-DL can represent E-R models!.....	6
5.3 Using OWL-DL to build the TMDM.....	6
5.3.1 The Approach In a Nutshell.....	6
5.3.2 Why would Topic Map authors want to use OWL-DL to build TMs?.....	6
6 Current Status.....	7
6.1 Work conducted to date.....	7
6.2 Functionality Implemented and Outstanding.....	7
6.3 Intention.....	7
7 Designing and Building the TMDM Ontology.....	7
7.1 The Design Phase.....	7
7.1.1 Design Choices .....	8
7.1.2 Identifying Individuals.....	8
7.1.3 Filling in the rest of the Model.....	8
7.1.4 A note about the use of Topics for Typing other Topic Map Objects.....	8
7.2 The Ontology Model.....	8
7.3 Additional Constraints Implemented.....	9
7.3.1 Internal and External Occurrences.....	10

7.3.2 At least one Locator for each Topic.....	10
7.3.3 Variant Name Scope must be a true Superset of Topic Name Scope .....	10
7.4 Implementation.....	11
8 Examples.....	11
8.1 Topics and Subjects.....	11
8.2 Occurrences.....	13
8.3 Topic Names and Variant Names.....	14
8.4 Associations.....	15
9 OWL capabilities for TM authoring.....	17
9.1 Consistency Checks and Automatic Inferencing.....	17
9.2 Additional Constraints.....	17
9.3 Querying.....	17
9.4 Visualisation.....	20
10 Outstanding Issues.....	22
10.1 The OWL-Full approach.....	22
10.2 The "Quick Fix" Workaround.....	22
10.3 The Extra Java code approach.....	22
10.3.1 Using Object Properties.....	22
10.3.2 Generating Instances of Association and Association Roles.....	23
11 Conclusions.....	23
12 Appendix A.....	23
Bibliography.....	28
The Author.....	29



# Building Topic Maps in OWL-DL

Anne Cregan

## § 1 Motivation

Anne Cregan, the author of this paper, is a PhD student enrolled at the University of New South Wales, and endorsed by the National Information and Communications Technology Australia (NICTA) Centre of Excellence. Part of NICTA's mandate is to assist Australian industry to achieve best practice in the use of IT. In mid-2004, Australian industry representatives from the software and analytics industries requested NICTA's assistance in understanding the relationship between ISO Topic Maps and the W3C's Semantic Web Recommendations, and exploring possibilities for interoperability. As part of this effort, Anne was asked to investigate the possibility of defining a mapping between the two standards. Being within NICTA's Knowledge Representation and Reasoning Program, which has a core capability in the use of formal logics for reasoning, she was encouraged to consider formal semantics in assessing possible solutions. This effort fits within the broader NICTA-wide priority challenge "From Data to Knowledge".

## § 2 Scope of the Approach

This paper presents a mapping and construction of the TMDM model as an OWL-DL ontology, as a proof-of-concept of the viability of building Topic Maps based on the TMDM directly in OWL-DL. This approach gives TM authors access to OWL-DL's capabilities and tools for expressing constraints, doing constraint checking and automated reasoning, and for the querying and visualisation of ontologies. There is significant overlap with the capabilities currently provided by Topic Map engines and envisaged to be provided by the Topic Map Constraint and Query Languages in development.

The approach presented herein should not be considered to be a final or definitive solution for TMDM/OWL interoperability. The TMDM itself is still officially at Draft status and in the process of being finalised. One aspect of the specified required functionality of the TMDM is not fully implemented in this construction (see Section 10 Outstanding Issues), and the remainder has been subjected only to rudimentary testing.

Furthermore, this approach has no status as a Standard or Recommendation with either the ISO or the W3C, although both bodies have indicated receptiveness to the concept and a willingness to work on interoperability, and the ISO Topic Map Working Group has kindly reviewed an earlier proposal outlining the approach, and indicated its willingness to provide continued assistance in providing feedback on OWL representations for the TMDM.

By virtue of being implementation as an OWL-DL ontology, a Description Logic Formal Semantics is provided but the specifics of the Semantics are not detailed herein, and interested parties would need to consult the relevant OWL documentation.

Whilst OWL and its tools provide Querying and Constraint capabilities which significantly overlap with the requirements specified for a Topic Map Query Language and Constraint Language, it is not claimed that they necessarily meet all such requirements.

The TMDM ontology and examples as built using Protege are freely available at <http://www.cse.unsw.edu.au/~annec/> to anyone who may wish to download them for closer inspection and evaluation, and feedback via email to [annec@cse.unsw.edu.au](mailto:annec@cse.unsw.edu.au) is welcomed.

## § 3 Background

### 3.1 The ISO's Topic Map Tradition

#### 3.1.1 Topic Map Origins

Topic Maps are an ISO/IEC standard [TM] (ISO 13250-1) for mapping both web and real-world information resources, by reifying real-world resources as "subjects", and creating "topic" constructs to capture their characteristics and relationships with other topics and subjects. Dubbed the "GPS of the information universe", they are akin to an electronic back-of-book index, facilitating information navigation and retrieval in on-line environments. In 2001, the first two standard syntaxes for Topic Maps were published by the ISO: XTM1.0 in XML [XTM1.0] and another in HyTM.

### 3.1.2 Incomplete Specification of Topic Maps

Although there were non-trivial differences between the two Topic Map syntaxes, the original standard did not explain the relationship between them, nor did it make the underlying data model explicit. Additionally, both syntaxes failed to specify what Topic Map implementations were required to do in certain situations, and neither supported constraints such as: "A person must be born in a place" or "A person must have at least one name". Requirements subsequently drafted by the ISO Topic Map Working Group (WG3) for a Topic Map Constraint Language [TMCL] and a Topic Map Query Language [TMQL] both needed a clear description as to how Topic Map constructs are to be evaluated.

### 3.1.3 A Formal Data Model for Topic Maps

To address these issues, the WG3 commenced work on a Topic Map Data Model in May 2001, to date culminating in the official Draft ISO 13250-2 TMDM [TMDM] published in January 2005, which clarifies the representation and gives Topic Maps a formal data model for the first time, in Entity-Relationship model form. The TMDM document is currently (as at June 2005) at draft status, but close to its full completed form, whilst work on the specification of TMCL and TMQL standards, based on the Draft TMDM document, is progressing rapidly.

### 3.1.4 The TMDM is not the last word on Topic Maps

The reader is asked to note that within the ISO Topic Map Working Group, the TMDM is not considered to be definitive of all forms of Topic Maps as described by the ISO 13250-1 Standard, and the question of distilling the essence of the Topic Map paradigm and giving its formal representation is being worked on at a more abstract level, with the Topic Map Reference Model [TMRM] effort leading the charge. However, as this work is still maturing, this paper limits its scope to the consideration of Topic Maps as they are described by the TMDM.

## 3.2 The Semantic Web Initiative

### 3.2.1 Berners-Lee's Vision

The Semantic Web is the vision of Tim Berners-Lee, and is a World-Wide Web Consortium (W3C)-led initiative with the goal of providing technologies and standards for semantic markup of online information resources, enabling improved web navigation and supporting intelligent web services. It is an XML-based technology, using Resource Description Framework [RDF] and Web Ontology Language [OWL] layers superimposed on XML to provide more expressive representations of the characteristics of, and relationships between, logical entities. Further layers and functionalities are planned as part of the overall vision.

### 3.2.2 Formal Semantics

The original RDF Recommendations did not specify a formal semantics, but later work has provided this, and OWL has built on to this further. OWL-DL, a formal W3C recommendation finalised in 2004, is a subset of the full OWL language which has a Description Logic (DL) equivalent semantics, providing decidable reasoning. Tools to support the generation and use of OWL-DL ontologies are well-developed, notably including Stanford University's Protégé [Protege] package, which provides a GUI that generates OWL code and gives access to modules for ontology visualisation, querying, consistency checking and automated reasoning.

## § 4 Interoperability Issues

### 4.1 Topic Maps and RDF

#### 4.1.1 Interoperability Challenges

The ISO and W3C initiatives were originally developed independently, sheerly through lack of awareness of each other's work. On discovery of this around 2001/2002, many felt that as both the Topic Map and Semantic Web standards provide the means to construct meta-level maps of information entities, there must be synergies to be exploited. However, previous attempts at achieving interoperability between Topic Maps and RDF have been problematic for the following reasons, amongst others:

- Neither Topic Maps nor RDF had stable formalized data models at the time
- Topic Maps use n-ary relations to express associations between topics, whilst RDF is composed of triples which express binary relations.

- Topic Maps use two kinds of locators: both direct and indirect, whilst RDF has only one kind of locator mechanism.
- The concept of scoping, which is fundamental to the Topic Map paradigm, has no obvious equivalent in RDF.
- The use of Topics as Types within Topic Maps has been problematic — should Topics be interpreted as classes, instances, or both? How does one retain the flexibility that the Topic Map Paradigm offers in this respect, where any Topic can take on a multiplicity of uses, whilst still providing an unambiguous formal model for representation?

It has proved to be a complex and non-trivial issue which does not easily lend itself to a general solution appropriate for every case. A full analysis is available in a survey of RDF/Topic Map Interoperability Proposals [RDFTM], recently published by the W3C's RDF/Topic Maps Interoperability Task Force. This task force, which is part of the W3C's Best Practice initiative, includes representatives from both the ISO Topic Map Working and the W3C, and is currently working to develop a standard for interoperability between RDF and Topic Maps.

## 4.2 TMDM and OWL

### 4.2.1 New Interoperability Possibilities

Many of the problems in achieving interoperability between RDF and Topic Maps appear to stem from RDF being insufficiently expressive to capture the full semantics of TM constructs in a straightforward way. The lack of a formal TM specification prior to the TMDM coming into being has also been a considerable hindrance. So the release of an official ISO Draft of the TMDM, combined with the advent of the more expressive OWL language as an official W3C Recommendation, is potentially of huge benefit for interoperability efforts, and begs an investigation of the fit between the two.

### 4.2.2 TMDM to OWL?

The TMDM is fundamentally a data model, whilst OWL is a language. It therefore seems natural to investigate whether the model defined by the TMDM can be represented and implemented using the semantics provided by the OWL language. That investigation is the primary focus of this work. The mapping produced is an Object Mapping as opposed to a Semantic Mapping, in that it takes the letter of the TMDM specification as bible and represents it fully in OWL, and produces a result where core TMDM constructs are mapped not to core OWL constructs, but to an OWL ontology built using them.

Topic Maps are then authored in OWL simply by importing the TMDM ontology and populating the ontology with instances. If constructed in this way, such Topic Maps are quite intuitive to build, and are directly translatable to other Topic Map languages based on the TMDM.

Although not covered here, there is also the possibility for another OWL representation for Topic Maps, also being considered by the author: one which does not adhere to the letter of the TMDM but produces the desired functionality using a simpler, more natural OWL ontology.

### 4.2.3 OWL to TMDM?

The question of whether ontologies written in OWL in a more general sense can be mapped to Topic Map languages or models based on the TMDM is also very worthy of consideration, but unfortunately cannot yet be fully investigated. It would require a complete specification of the TMCL, as OWL clearly provides the means to express constraints that are beyond the scope of the TMDM's expressivity, and fall squarely within the ken of the associated TMCL standard. The TMCL specification is taking shape rapidly, whilst the TMRM in development may also inform the issue, but such an investigation is still premature at the time of writing.

## § 5 TMDM to OWL Mapping Investigation

### 5.1 The TMDM side

#### 5.1.1 What does the TMDM give us?

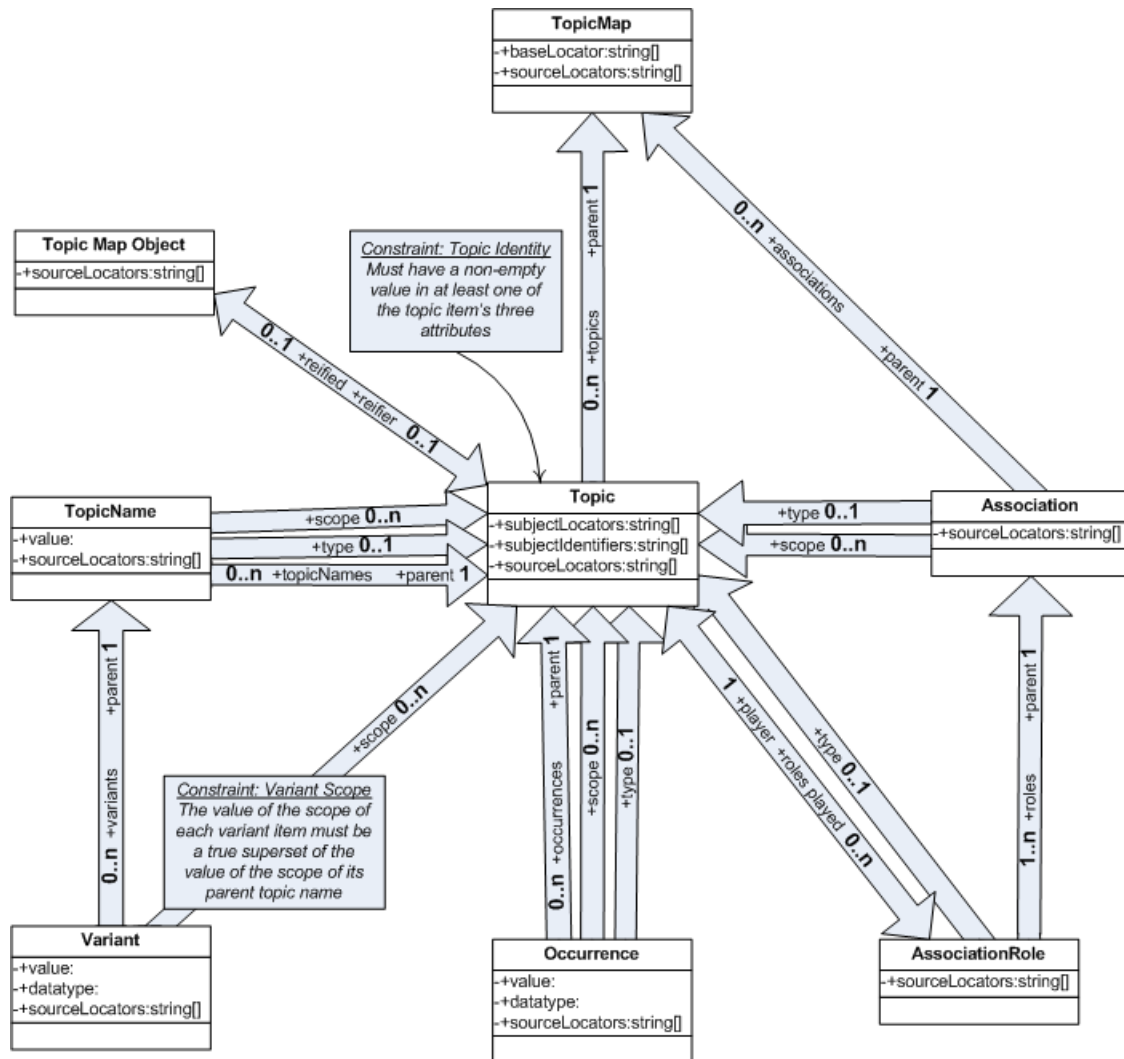
The drafted TMDM offers some significant benefits over previous Topic Map specifications. The majority of it is represented as Entity-Relationship diagrams with explanatory text, and the clear delineation of Topic Map constructs into Entities, Relationships and Attributes is a significant and most welcome contribution. The directions and cardinalities of relationships between Topic Map entities are clearly



represented. Type and parent relationships are stated separately and explicitly. The different kinds of relationships Topics may have with Subjects are clearly represented. Data attributes are given for the logical entities, which include values, data types and locators such as URI references. The draft also introduces two additional constraints to ensure the integrity of Topic Maps based on the model.

For ease of reference, all the E-R diagrams provided by the TMDM specification plus the two additional constraints have been collated into a single diagram shown at Figure 1. Note that "Topic Map Object" can be any of the other Topic Map Objects shown (Association, Occurrence, Topic, etc).

Figure 1



The TMDM Model, showing collated E-R diagram and two additional constraints

### 5.1.2 But wait, lurking in the back of the TMDM closet

However, lurking in the last section of the TMDM specification (at least in the current draft) under the rather innocuous title of "Published Subjects", are hidden two of the most fundamental Topic Map constructs: "Type-Instance" and "SuperType-SubType". In this section they are specified to be special instances of Associations with specific types between role-playing Topics which play special instances of Association Roles with specific types. The types are grounded by being set to specified Published Subject Indicators for "Type-Instance" and "SuperType-SubType". This is obviously much heavier and

more complex than standard ontological Class-Instance and SuperClass-SubClass relations. In terms of the Data Model given up to this point, they are able to be interpreted simply as special instances of its constructs; rather unwieldy and convoluted to be sure, but not particularly problematic.

However, unlike any other instances of Associations in the TMDM, instances of these special types of Associations are required to have the functionality of ontological Class-Instance and SuperClass-SubClass relations, in that instances of "SuperType-SubType" Associations must be transitive, and instances of "Type-Instance" Associations, although not transitive in themselves, must ensure that any Instances of a Type are also Instances of any SuperTypes of that Type. That is, if A is an Instance of Type B, and B is a SubType of SuperType C, then A must be an Instance of C also.

On consideration, this seems to entail that Topic Map processing is required to create new instances of Associations and Association Roles on the fly in order to capture the flow-on effects implied here. Whilst this is rather inconvenient, it is not necessarily insurmountable for an OWL construction if a little extra Java code is attached. This issue is put aside for the moment and revisited in detail in Section 10 Outstanding Issues.

## 5.2 The OWL side

OWL is the most recently developed, and most expressive layer of the W3C's Semantic Web Recommendations to date. It is a Web Ontology Language, designed for the representation of classes containing individuals which have relations with each other and data attributes. It also supports the expression of constraints on these entities.

### 5.2.1 Flavours of OWL

There are currently three "flavours" of OWL which vary in their level of expressivity. OWL-Lite has the least expressivity and is a subset of OWL-DL. Both OWL-Lite and OWL-DL have a Description Logic equivalent semantics. Description Logics are a decidable fragment of First Order Logic. Being decidable means that it is possible to ask whether certain things are the case and be guaranteed of getting an answer in a finite number of steps. For Description Logics, the kinds of things that can be requested are:

- computing the inferred superclasses of a class
- determining whether or not a class is consistent (a class is inconsistent if it cannot possible have any instances)
- deciding whether or not one class is a subclass of another class

Automated Description Logic reasoners are available to perform these inferencing services and others for OWL-DL and OWL-Lite.

The full OWL language, OWL-Full, supports additional expressivity beyond OWL-DL and OWL-Lite, such as allowing a thing to be both an individual and a class at the same time, but sacrifices the decidability of reasoning. This is largely due to the ability OWL-Full has to create self-referential loops: a class is allowed to contain itself as an instance. OWL-Full does not have a Description Logic semantics, and while it is certainly still possible theoretically to perform many inferencing tasks effectively on OWL-Full, currently the majority of OWL tools and reasoners do not support OWL-Full, whilst they provide good support for the OWL-DL and OWL-Lite sublanguages.

### 5.2.2 Which flavour of OWL tastes best to make Topic Maps?

Whilst OWL-Full is a more natural medium for expressing the notion that Topics may be both Classes and Instances, if one's aim is to build a practical, functioning ontology with full support for querying and constraint checking, at this time OWL-DL is the preferred OWL sublanguage to use, as it gives the highest level of expressivity whilst still being decidable, having a well-defined semantics and being well-supported by OWL tools.

However, OWL-DL may not always be the flavour of choice in future. Since the main purpose of Topic Maps is to enable humans to find things, rather than machines to reason about them, and OWL-Full's looser expressivity is a better fit for this, it may not ultimately be an issue that OWL-Full is undecidable, as long as there are functioning tools to provide the kind of consistency checks and querying needed by Topic Maps.

For effective navigation, most of this needs only to be done at a local scale - that is, looking only at constructs within a local area of the TM, and how they are related to their neighbouring constructs. It should be possible to do this in most cases without coming up against the issue of undecidability. As an

analogy, consider one of M.C.Escher's "strange-loop" drawings like "Ascending and Descending" which may be found at <http://www.mcescher.com/indexuk.html> : If you look at only a localised part of the picture it looks perfectly sensible, even though when you look at the whole picture, you realise there's an impossible loop. Because of the way the human mind works, it may be argued that it is perfectly reasonable to implement such loops in Topic Maps, because they reflect the way humans think about information, and having loops in Topic Maps doesn't stop them from being useful maps showing where things are and how they relate to each other. In fact, the loop can be considered to be a valid part of the map!

### 5.2.3 What can be asserted in OWL-DL?

In OWL-DL, Classes and Individuals need to be kept separate: no thing can be both a Class and an Instance of a Class. Classes may be related via a SuperClass-SubClass hierarchy, and may also be declared to be disjoint from each other, that is, to have no overlapping Instances. Individuals may be asserted to be Instances of Classes, and may be related to each other via OWL's Object Properties. They may also be given data attributes, using OWL's Datatype Properties. Both Object and Datatype Properties may be constrained on domain, range, and cardinality, and given various properties such as transitivity and symmetry. Properties may be constrained not only globally, but with respect to specific Classes. OWL Object Properties are one-directional, but may be linked to a corresponding property in the opposite direction using OWL's built-in "inverseOf" relation construct.

### 5.2.4 OWL-DL can represent E-R models!

The commonality between OWL-DL's expressiveness and standard Entity-Relationship models renders OWL capable of representing a simple Entity-Relationship model by the following method:

- Entities are mapped to OWL Classes
- Instances of Entities are mapped to OWL Individuals belonging to those OWL Classes
- Relationships between Entities are mapped to OWL Object Properties linking OWL Classes
 

The relationship's cardinality may be captured within the Object Property, as may its directionality, through domain and range specification. To represent bi-directional relations, an additional Object Property in the opposite direction is created and declared to be an inverse property of the original.
- Attributes of Entities are mapped to OWL Datatype Properties linking OWL Classes to Data Value Ranges. Attribute Values of the Instances of Entities correspond to specific values from the Data Value Ranges for individual within the OWL class that corresponds to the Entity. These Properties may also have explicit cardinality.

It is not claimed that such a mapping is the only possible way one might map an E-R diagram to OWL-DL; merely that it is a reasonable, natural and straightforward approach. The question of mapping UML (the means to represent E-R diagrams) to OWL is a research issue in its own right.

## 5.3 Using OWL-DL to build the TMDM

### 5.3.1 The Approach In a Nutshell

The method above gives a straightforward algorithm for mapping simple E-R diagrams to OWL-DL. As the TMDM is largely specified as such a simple E-R model (refer to Figure 1), those parts of it can be directly deposited into OWL-DL using the method given above. The TMDM also has a couple of extra constraints to do with cardinality of relations, and whilst these take a little more thought, they may also be implemented in OWL-DL. With the TMDM constructs now constructed in OWL-DL, Topic Map authors may then build their own Topic Maps in OWL-DL simply by importing the OWL ontology that defines these constructs, and populating it with instances.

### 5.3.2 Why would Topic Map authors want to use OWL-DL to build TMs?

The advantage of using OWL-DL for Topic Map authoring based on the TMDM ontology is that the resulting Topic Maps are guaranteed to conform with the TMDM, as all the conditions for conformance are implemented within the TMDM ontology and thus passed on to the Topic Map ontologies based on it. The author can also use OWL's expressivity to express additional constraints on their Topic Maps in a way that closely corresponds with the desired functionality of a TMCL, and can use the existing, freely available OWL tools to query their Topic Maps in a way that closely corresponds with TMQL functionality.

By virtue of using OWL-DL, the Topic Map author gains the benefits of:

- Access to a well-developed suite of free, open-source APIs, visualisation, querying and reasoning tools : For instance, Protege combined with the RACER [RACER] reasoner provide the means for both querying Topic Maps and enforcing constraints, meeting many of the ISO's requirements for a Topic Map Query Language and a Topic Map Constraint Language
- A clearly defined Description Logic formal semantics for the resulting Topic Maps. Semantic Web reasoning services and Intelligent Agent services are likely to become increasingly sophisticated and capable, and having a formal semantics helps open the door to these.

Whilst OWL tools do not currently include an engine specifically for the navigation of Topic Maps, the resulting Topic Maps would be directly translatable to any other TMDM-based Topic Map languages which become available, as the representation is equivalent. If the method is widely accepted, it is likely that a means to import Topic Maps written in OWL to commercially available Topic Map engines would be made available, providing TM authors with all the advantages of both worlds. By the same token, it would also be possible to port Topic Maps written in other TMDM-based TM languages into OWL.

## § 6 Current Status

### 6.1 Work conducted to date

Using the method described above, the TMDM E-R model shown at Figure 1 was translated into an OWL-DL ontology summarised at Figure 2, and implemented using Protege. Some sample Topic Maps were built as OWL-DL ontologies in Protege by importing the TMDM ontology to define the Topic Map constructs and populating it with instances. The construction in Protege is able to successfully utilise Protege's querying plugin and RACER reasoning plugin for constraint checking, as well as the other plugins that Protege provides, including those for Ontology visualisation.

The OWL-DL construction for the TMDM was submitted to the ISO TMDM authors for review and comment, and discussed with the WG3 at meetings held in May 2005. They agreed it was indeed a faithful representation of the TMDM, implemented at an object level.

### 6.2 Functionality Implemented and Outstanding

Note that the whilst the TMDM's "Type-Instance" and "Supertype-Subtype" relationships as described in Section 7 of the TMDM Draft document may be fully asserted as instances of the ontology built here, the flow-on functionality is not currently implemented: that is, the transitivity of instances of Supertype-Subtype Associations, and the upward inheritance of Instances within Type-Instance associations to the SuperTypes of a Type expressed in Supertype-Subtype relations. Some approaches for implementing this functionality are discussed in Section 10 Outstanding Issues. With this exception, all other desired functionality required by the TMDM is believed to be captured in the ontology presented in the following section.

### 6.3 Intention

If the outstanding issue can be resolved satisfactorily, and the resulting OWL syntax accepted by the relevant Standards Bodies as a valid syntax for the TMDM, it would then be made available online at an appropriate location, for Topic Map authors to import to OWL. As described, by providing the TMDM constructs for authoring Topic Maps in OWL-DL, users are enabled to author Topic Maps in OWL-DL which are TMDM compliant. This method for Topic Map construction has no reliance on XTM or any other TM languages: no XTM references are necessary, either in the TMDM ontology or in the user-defined Topic Map ontologies. It may be reasonably expected though, that if an OWL syntax were accepted as a standard for authoring Topic Maps, then Topic Map engines and tools would develop the ability to import Topic Maps written in this syntax, as they are doing with RDF.

## § 7 Designing and Building the TMDM Ontology

The following sections describe the design and construction of the TMDM as an OWL-DL ontology in detail.

### 7.1 The Design Phase

The method used corresponds to that described above for mapping Entity-Relationship Models into OWL-DL, but is here explained at a micro level with specific reference to the TMDM model shown at Figure 1. Visually comparing Figure 1 and Figure 2 should also aid the reader's understanding.

### 7.1.1 Design Choices

As with the construction of any OWL ontology, some design choices must be made. For reasons of implementability discussed above, it is desirable, at least currently, to stay within the expressivity of OWL-DL rather than to use OWL-Full, and to enable this, the TMDM ontology needs to maintain a clear distinction between Classes and Individuals.

### 7.1.2 Identifying Individuals

The first task is to identify the nature of the individuals in user Topic Maps, keeping in mind that whilst OWL's Object and Data Properties are defined using Classes as their domains and ranges, the properties ultimately link the Individuals within those Classes. That is, if we want to be able to assert that something has Object and Datatype Properties with particular values, then it needs to be an Individual, not a Class. Accordingly, the candidates for Individuals are taken to be the individual Topics, Occurrences, Associations and so on that Topic Map authors would ultimately want to define relationships between, and characteristics for.

### 7.1.3 Filling in the rest of the Model

Having decided that these are to be the Individuals, the classes "Topic", "Occurrence" and "Association" and so on are created to contain these individuals. These are subsequently used as domains and ranges to define OWL Object Properties between individuals, providing the general framework defining which groups of Individuals may be related to which other groups of Individuals, and in what way (cardinality of relations, etc). The Topic Map author will ultimately import these these imported Classes and Properties and instantiate them with her own Topics, Associations, linkages between them and so on to build her own Topic Maps.

The attributes of each entity in Figure 1 (source locator, etc) are translated to OWL Datatype Properties belonging to Individuals within the OWL Classes. Constraints may be implemented as restrictions on the OWL classes.

### 7.1.4 A note about the use of Topics for Typing other Topic Map Objects

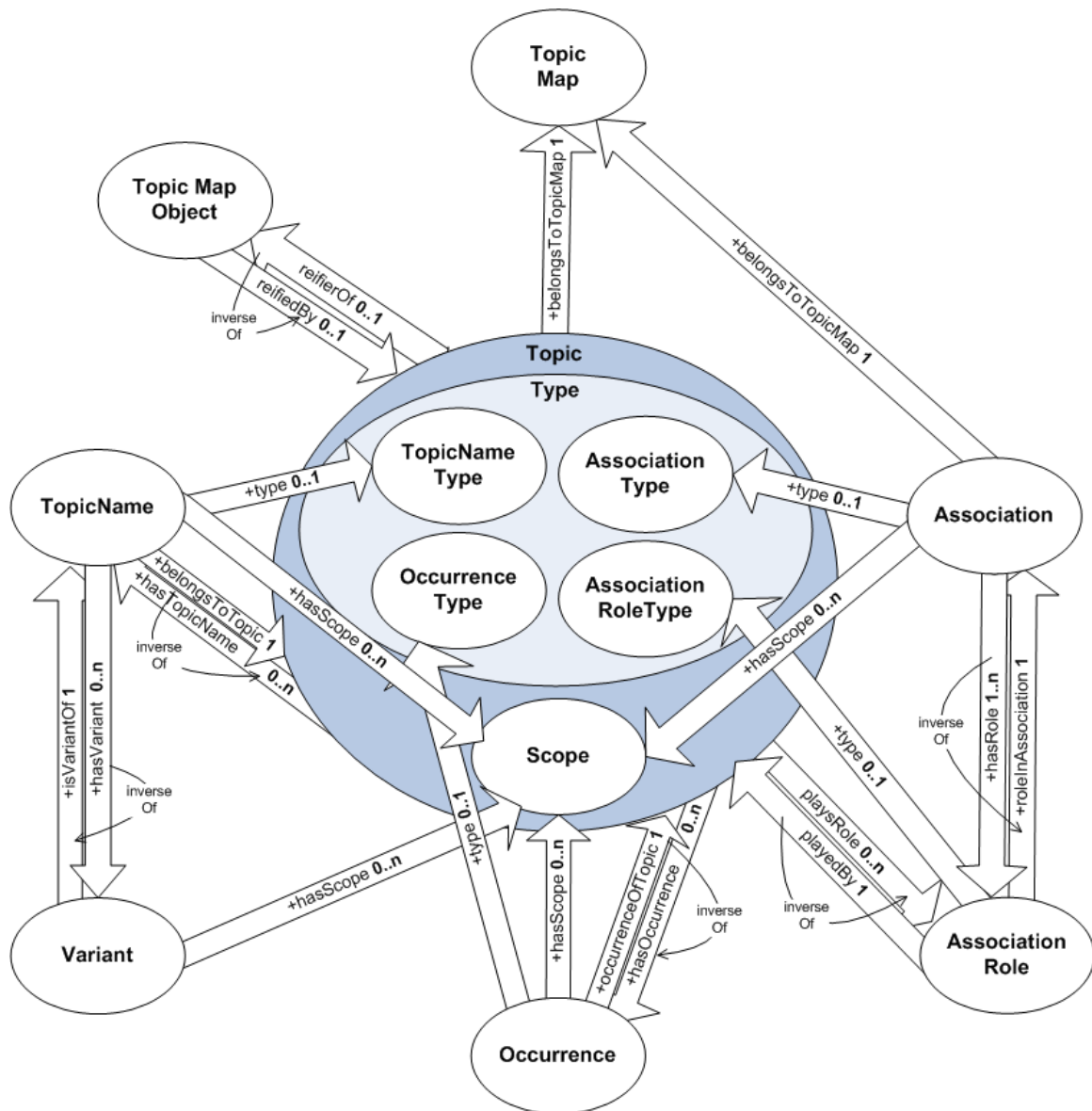
A distinguishing feature of Topic Maps is that Topics are used as Types for other Topic Map Objects, whilst also being used in all the ways that regular Topics are. Normally in an ontology, a type would be represented as an OWL Class, and Individuals would be typed by being asserted to belong to the Class. However, if this were to be done here, Topics would need to be represented as both OWL Classes and OWL Individuals, as they need to be able to be Types whilst still having Objects and Datatype Property values in the way individuals do. This would require the use of OWL-Full.

However, as OWL-DL provides the expressivity to define complex relations between individuals using Object Properties, it is preferable to represent Types as individual Topics, and to represent the "type" of Topic Map Objects, not through class membership, but as an Object Property between the Topic Map Object and the Topic which types it. The issue of Typing for Topics themselves is a more complex issue addressed at Section 10 Outstanding Issues.

## 7.2 The Ontology Model

The OWL-DL model for the TMDM used for construction is presented at Figure 2. As there is not yet a standard graphical notation for representing OWL ontologies, RDF graph notation has been extended: Classes are represented as ovals; the nesting of an oval within a larger oval indicates a subclass-superclass relation; arrows represent object properties, labelled with the Object Property's name and cardinality, and indicating direction by pointing from Domain to Range. To avoid cluttering the diagram, Datatype Properties and complex constraints are omitted, SuperProperty constructions are not shown and not all Inverse Properties are shown. Note that there are no Individuals in the TMDM ontology: these are defined in the process of Topic Map authoring. The TMDM ontology can be considered to be a template or mould of exactly the right dimensions for TM authors to fill up with instances, turning out perfectly formed Topic Maps.

Figure 2



The TMDM Ontology Model, showing Classes and Object Properties

Note the correspondences between Figures 1 and 2: Each entity (box) in Figure 1 has been translated to a class (oval) within Figure 2. Relationships between the entities in Figure 1 are captured as Object Properties in Figure 2. Where there is a bi-directional relation in Figure 1, two relations have been given in Figure 2, related as inverses. For additional ease of use, many additional inverse relations have also been defined. Logically, these are already implied by the TMDM, and by making the inverse forms available, users are given more flexibility to write their own Topic Map ontologies in OWL in the most convenient way, as they can start from either end.

### 7.3 Additional Constraints Implemented

The Ontology described above and shown in Figure 2 was built in OWL-DL using Stanford University's Protégé package for Ontology Engineering. In order to achieve all the required functionality specified by



the TMDM, some additional constraints were needed that are worthy of additional commentary. These are described below.

### 7.3.1 Internal and External Occurrences

Although Topics may have any number of Occurrences, each Occurrence may have only one value, but it may be either a string (indicating an internal Occurrence) or a URI (indicating an external Occurrence).

Correspondingly, two different Datatype Properties were attached to Occurrences:

- “occurrenceTextString” which maps to a string value, used for asserting internal occurrences and
- “occurrenceTextLocatedAt” which maps to a URI value, used for asserting external occurrences.

By using the appropriate Datatype Property, one may distinguish between internal and external Occurrences and map each to the appropriate Data Type. However, both are defined as subproperties of an “OccurrenceText” Datatype Property, which is defined as a functional property (that is, it may only map to one Data Value), ensuring that there is only ever one Data Value (which may be either a string or a URI) for each instance of the class Occurrence.

A similar situation for Variants applies, dealt with using a “variantName” superproperty for “variantNameString” “VariantNameLocatedAt” Datatype Properties.

### 7.3.2 At least one Locator for each Topic

According to the mapping described previously, the three locators relevant to Topics were mapped as Datatype Properties with domains as follows:

- SourceLocator: Topic Map Object
- SubjectIdentifier: Topic
- SubjectIndicator: Topic

According to the constraint specified by the TMDM, it is necessary to ensure that for all Topics, at least one of these locators had a value. However, the non-empty locator can be of any of the three kinds available for Topics.

To achieve this functionality, a superproperty Datatype Property called “Locator” containing these three Datatype Properties was created. Then a restriction was placed on the class of Topics with respect to the “Locator” Datatype Property, requiring it to have a minimum cardinality of 1. This ensures that for all Topics, at least one of the three locator sub-Properties MUST be non-empty. As the restriction only applies to the class “Topic”, not to all classes in the domains of the Properties, it does not affect other TM objects using the “Source Locator” Datatype Property.

### 7.3.3 Variant Name Scope must be a true Superset of Topic Name Scope

The TMDM places the following constraint on Variant Scopes: the set of scopes of each Variant of a Topic Name must be a true superset of the set of scopes of its parent Topic Name. This can be interpreted to mean that a Variant name must inherit all the scopes of its parent Topic Name plus have at least one of its own in addition to this.

According to the methods previously described, an Object Property “isVariantOf” was defined with domain “Variant” and range “TopicName”, and with a cardinality of 1, as a variant must belong to exactly one parent Topic Name. Additionally, the “hasScope” Object Property, of unlimited cardinality, relates both Variant and Topic Name to the class “Scope” : a subclass of “Topic” containing those Topics being used as Scopes.

By making “isVariantOf” a subproperty of “hasScope”, and including “TopicName” in the range of “hasScope”, a variant may be considered to be scoped by its Parent Topic Name. By then making the “hasScope” property transitive, the Variant inherits the scopes of its parent Topic Name. To ensure that Variant will have a least one scope relating it directly to the class “Scope”, another subproperty of “hasScope” is defined with has domain “Variant” and range “Scope”, and a minimum cardinality of 1 is placed on this property. Voila!

To tidy up, as only variants are allowed to be scoped by Topic Names, and other Topic Map Objects are not, we limit the hasScope Property on the other classes which use it to only allow values from the class “Scope”, and not from “TopicName”.

## 7.4 Implementation

The OWL-DL code generated by Protege for the TMDM ontology described above is included at the end of this paper and is also available at the author's website, along with the Protege ontology itself and examples.

## § 8 Examples

In this section, the OWL TMDM constructs defined above are used to author an example Topic Map in OWL-DL which describes an Italian Opera domain. The example ontology's header (not shown) sets up namespaces, makes the standard XML, RDF and OWL inclusions, and imports the TMDM Ontology described above. Use of the prefix "tm" in the OWL code shown below indicates a construct is sourced from the imported TMDM ontology. The names used as identifiers for instances of Topic Map elements are completely arbitrary and up to the author to choose, the only requirement being that they must be unique within the authored Topic Map ontology.

### 8.1 Topics and Subjects

Say that within the Italian Opera Topic Map we would like to create a topic called "Puccini" to represent the composer Giacomo Puccini. This is done simply by:

- creating an instance of the class "Topic" from the TMDM ontology,
- calling it "Puccini",
- and giving its Subject Identifier Property an appropriate value to identify the subject that this Topic identifies: in this case the URL of an online wiki about the composer Giacomo Puccini.

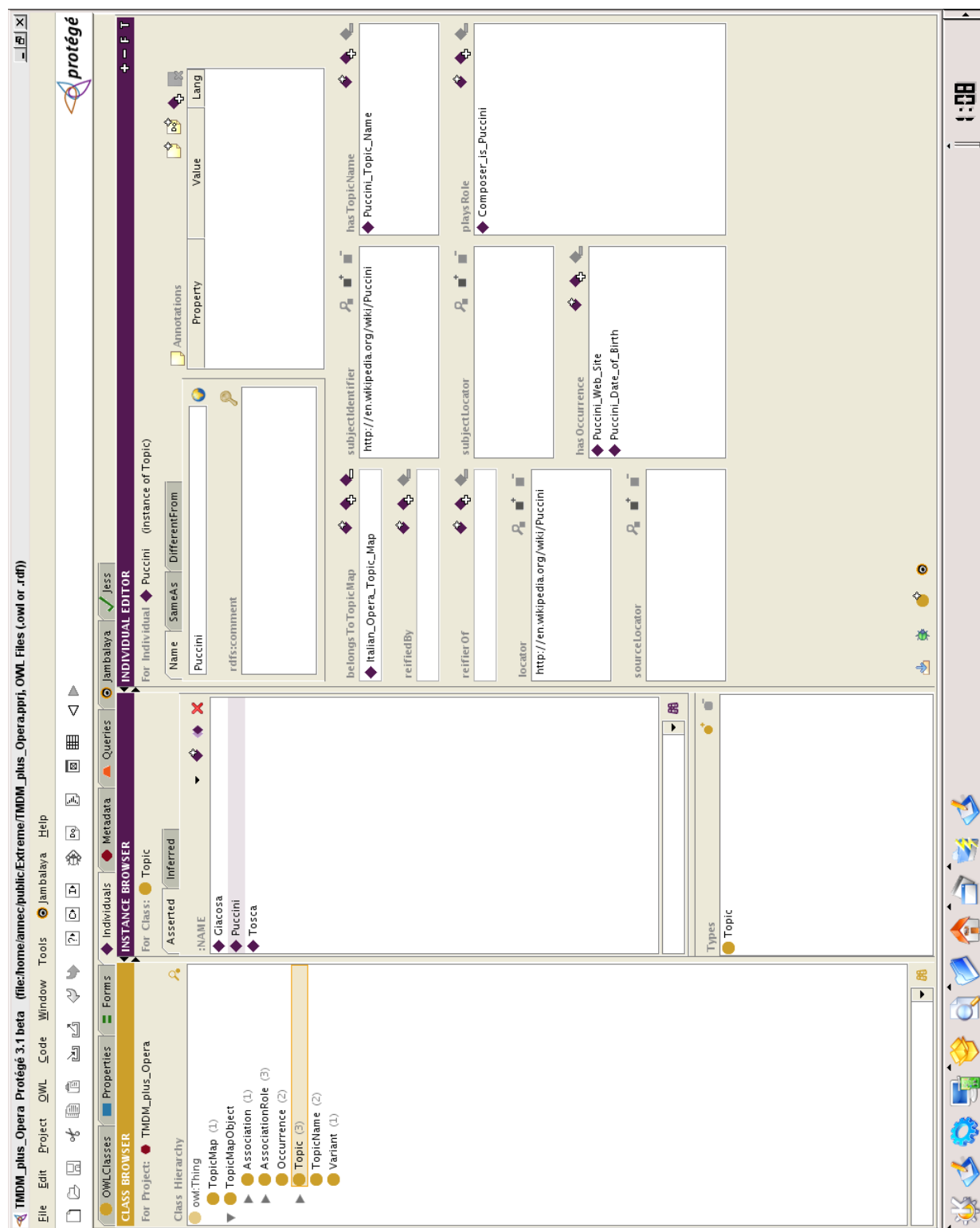
The corresponding OWL code is shown below:

```
<tm:Topic rdf:ID="Puccini">
  <tm:subjectIdentifier rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    http://en.wikipedia.org/wiki/Puccini</tm:subjectIdentifier>
</tm:Topic>
```

If using a GUI OWL tool such as Protégé, the OWL code is generated automatically by undertaking the actions described above. Protege generates an input form based on the Properties defined for the Individual, and the user can simply assign values to the Properties by select from previously defined entities, or enter new data using this form. A Protege screenshot used to assign values to the "Puccini" Topic is shown below.



Figure 3



Protege Screenshot : Entering Values for the Topic "Puccini" in the Italian Opera Topic Map

---

#### Processing Points to Note:

- The value given to the Topic's "Subject Identifier" property gives the Topic's "Locator" SuperProperty a value, thus satisfying the requirement that "Locator" has a minimum cardinality of 1. When running consistency checks, an error would be issued for every Topic does not have at least one value for "Locator".
- As the Topic is now defined, it can be used later in the OWL ontology by using the ID "Puccini" to refer to it ( `rdf:about = "Puccini"` )

### 8.2 Occurrences

We would now like to assert two occurrences for the topic "Puccini" in our Italian Opera Topic Map:

- Puccini's date of birth, which is a string, and
- a website where Puccini's operas can be found, which is a URL

This is done by:

1. Firstly, asserting that the Topic "Puccini" has an Occurrence called "Puccini\_Date\_Of\_Birth", where:
  - The "type" property of the Occurrence is an instance of the class "Occurrence Type", called "Date\_of\_Birth"
  - The "occurrenceTextString" property of the Occurrence is the string "1858-12-22" (Puccini's birthdate)
2. A second assertion that the Topic "Puccini" has another Occurrence called "Puccini\_Web\_Site", where:
  - The "type" property of the Occurrence is an instance of the class "Occurrence Type", called "Web\_Site"
  - The "occurrenceLocatedAt" property of the Occurrence is the URL "http://www.puccini.it/"

The corresponding OWL code is:

```
<tm:Topic rdf:about="Puccini">
  <tm:hasOccurrence>
    <tm:Occurrence rdf:ID="Puccini_Date_of_Birth">
      <tm:type>
        <tm:OccurrenceType rdf:ID="Date_of_Birth"/>
      </tm:type>
      <tm:occurrenceTextString rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#string">1858-12-22
      </tm:occurrenceTextString>
    </tm:Occurrence>
  </tm:hasOccurrence>

  <tm:hasOccurrence>
    <tm:Occurrence rdf:ID="Puccini_Web_Site">
      <tm:type>
        <tm:OccurrenceType rdf:ID="Web_site"/>
      </tm:type>
      <tm:occurrenceTextLocatedAt rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#string">http://www.puccini.it/
      </tm:occurrenceTextLocatedAt>
    </tm:Occurrence>
  </tm:hasOccurrence>
</tm:Topic>
```

#### Processing Points to Note:

- Consistency checks ensure that each occurrence has only one data value for "occurrenceText", although it can be either a string (when using "occurrenceTextString" subproperty) or a URI (when using "occurrenceTextLocatedAt" subproperty).

- Conversely, we may assert as many Occurrences as we like, or none at all for the Topic, since it has no cardinality restriction. Each such assertion links the instance of Topic to a separate, newly created instance of Occurrence.
- Instances of "OccurrenceType" are being generated in the code above within the "type" declaration for each Occurrence. Since Occurrence is a subclass of Topic, these are automatically also Topics, and once created, may be referred to and used in all the ways that Topics are, and inherit all the constraints and properties that Topics have.
- An Occurrence is only allowed to have at most one "type" property and it must be from the "OccurrenceType" subclass of Topic. If we try to give the Occurrence more than one value for its type or to link it to any other class, the consistency checks will fail.
- Because of the way the inverse properties are set up, every "hasOccurrence" asserted between a Topic and an Occurrence automatically generates an "isOccurrenceOf" property between the Occurrence and Topic in the opposite direction.

### 8.3 Topic Names and Variant Names

Say we would now like to give the "Puccini" Topic a Topic Name (also "Puccini") with scope of Normal\_Name, and an associated Variant Name which has the full name "Giacomo Puccini" with scope of Long\_Name.

To do this, we would simply:

- create an instance of TopicName called "Puccini\_Topic\_Name" with topicNameString set to "Puccini" and Scope of "Normal\_name".
- assert that this Topic Name has a Variant using the "hasVariant" Object Property, and create an instance of Variant which has "variantNameString" of "Giacomo Puccini", and Scope of "long\_name".

The corresponding OWL code is:

```
<tm:TopicName rdf:ID="Puccini_Topic_Name">
  <tm:belongsToTopic rdf:resource="#Puccini" />
  <tm:hasScope>
    <tm:Scope rdf:ID="Normal_name">
    </tm:Scope>
  </tm:hasScope>
  <tm:topicNameString rdf:datatype=
    "http://www.w3.org/2001/XMLSchema#string">Puccini
  </tm:topicNameString>

  <tm:hasVariant>
    <tm:Variant rdf:ID="Puccini_Long_Name">
      <tm:variantNameString rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#string">Giacomo Puccini
      </tm:variantNameString>
      <tm:variantScope>
        <tm:Scope rdf:ID="Long_name" />
      </tm:variantScope>
    </tm:Variant>
  </tm:hasVariant>
</tm:TopicName>
```

#### Processing Points to Note:

- When "belongsToTopic" is declared between an instance of Topic Name and an instance of Topic, the inverse property "hasTopicName" automatically links the Topic and the Topic Name in the opposite direction.
- A Topic Name is only allowed to belong to exactly one parent Topic, so if we tried to assert any more "belongsToTopic" properties linking it to other Topics, or if we did not assert any links to a parent Topic, the consistency checks would produce an error.
- Conversely, as there is no limit to the number of Topic Names a Topic may have, we are free to assert as many values of "hasTopicName" for each Topic as we like, or none at all.
- When "hasVariant" is declared between an instance of Topic Name and an instance of Variant, the inverse property "isVariantOf" automatically links the Topic and the Topic Name in the opposite direction

- A Variant is only allowed to belong to exactly one parent TopicName, so if we tried to assert any more “isVariantOf” properties linking it to other TopicNames, or if we did not assert any links to a parent TopicName, the consistency checks would produce an error.
- Conversely, as there is no limit to the number of Variants a Topic Name may have, we are free to assert as many values of “hasVariant” for each Topic Name as we like, or none at all.
- Instances of the class “Scope” are being generated within the code above within the “hasScope” declarations. Since Scope is a subclass of Topic, each instance of Scope is automatically a Topic, and once created, may be referred to and used in all the ways that Topics are and inherit all the constraints and properties that Topics.
- Through the transitivity of the “hasScope” property, which is a superproperty of “isVariantOf”, the Variant inherits all the scope elements of its parent Topic Name. After inferencing processes are completed, the Variant will be linked to Scopes which include both “Long\_name” and any scopes belonging to “Puccini\_Topic\_Name”; in this case “Normal\_name”. Thus the Variant will fulfil the constraint of having a true superset of the scope of its parent Topic Name. If we did not use the VariantScope property to link the Variant to at least one instance of Scope, this consistency check would fail.
- Although not shown in this example, TopicNames may be linked via the “type” property to at most one instance of TopicNameType. If we attempt to link via the type property more than once or to anything other than a TopicNameType, the consistency checks will fail. Variants are not allowed to have any type at all, and thus the “type” property does not include the class “Variant” as part of its domain. If we were to try to declare a “type” for an instance of “Variant”, consistency checks would indicate an error. When using a GUI such as Protégé, the input forms do not provide an option to select a “type” for an instance of Variant, as this class is not within the domain of the property.

## 8.4 Associations

Say we would now like to assert an association between topics Puccini, Giacosa and the opera Tosca, to indicate that Puccini and Giacosa wrote Tosca, with Puccini being the composer, Giacosa the librettist and Tosca the Opera.

This is done in the following way:

- Create an instance of the class “Association” and call it “Puccini\_and\_Giacosa\_Wrote\_Tosca”
- Using the “type” Object Property, link this instance of “Association” to an instance of “AssociationType” called “Written\_By”
- Using the “hasRole” Object Property, link the instance of “Association” to three newly created instances of “AssociationRole”:
  1. Firstly to an AssociationRole called “Librettist\_is\_Giacosa” which indicates that the Topic “Giacosa” plays the Role of “Librettist” in this association. This is achieved by linking the “type” property of the AssociationRole instance to an instance of AssociationRoleType called “Librettist”, and the “playedby” Property to a newly created instance of Topic called “Giacosa”
  2. Secondly to an AssociationRole called “Composer\_is\_Puccini” which indicates that the Topic “Puccini” plays the Role of “Composer” in this association. This is achieved by linking the “type” property of the AssociationRole instance to an instance of AssociationRoleType called “Composer” and the “playedby” Property to the previously created Topic instance called “Puccini”
  3. Thirdly to an AssociationRole called “Opera\_is\_Tosca” which indicates that the Topic “Tosca” plays the Role of “Opera” in this association. This is achieved by linking the “type” property of the AssociationRole instance to an instance of AssociationRoleType called “Opera”, and the “playedby” Property to a newly created instance of Topic called “Tosca”.

The corresponding OWL code is shown below:

```
<tm:Association rdf:ID="Puccini_and_Giacosa_Wrote_Tosca">
  <tm:type>
    <tm:AssociationType rdf:ID="Written_By">
    </tm:AssociationType>
```

```

</tm:type>

<tm:hasRole>
  <tm:AssociationRole rdf:ID="Librettist_is_Giacosa">
    <tm:type>
      <tm:AssociationRoleType rdf:ID="Librettist">
      </tm:AssociationRoleType>
    </tm:type>
    <tm:playedBy>
      <tm:Topic rdf:id="Giacosa">
      </tm:playedBy>
    </tm:AssociationRole>
  </tm:hasRole>

  <tm:hasRole>
    <tm:AssociationRole rdf:ID="Composer_is_Puccini">
      <tm:type>
        <tm:AssociationRoleType rdf:ID="Composer">
        </tm:AssociationRoleType>
      </tm:type>
      <tm:playedBy>
        <tm:Topic rdf:about="Puccini">
        </tm:playedBy>
      </tm:AssociationRole>
    </tm:hasRole>

  <tm:hasRole>
    <tm:AssociationRole rdf:ID="Opera_Is_Tosca">
      <tm:type>
        <tm:AssociationRoleType rdf:ID="Opera">
        </tm:AssociationRoleType>
      </tm:type>
      <tm:playedBy>
        <tm:Topic rdf:id="Tosca">
        </tm:playedBy>
      </tm:AssociationRole>
    </tm:hasRole>
  </tm:Association>

```

#### Processing Points to Note:

- When “hasRole” is declared between an instance of Association and an instance of AssociationRole, the inverse property “roleInAssociation” automatically links the instances of AssociationRole and Association in the opposite direction.
- An AssociationRole is only allowed to belong to exactly one parent Topic, so if we tried to assert any more “roleInAssociation” properties linking it to other Associations, or if we did not assert any links to a parent Association, the consistency checks would produce an error.
- Conversely, Associations must have at least one AssociationRole, so if we did not declare at least one link between this instance of Association and an instance of Association Role using either “hasRole” or its inverse somewhere within the ontology, consistency checks would fail.
- When “playedBy” is declared between an instance of AssociationRole and an instance of Topic, the inverse property “playsRole” automatically links the Topic and the AssociationRole in the opposite direction.
- “playedBy” must be linked to exactly one Topic and consistency checks will fail if this is not the case.
- Conversely, Topics may be linked to any number of AssociationRoles via the “playsRole” property.
- Each Association and AssociationRole may be linked by the “type” property to at most one instance of AssociationType and AssociationRoleType respectively. If we attempt to link via the “type” property more than once or to anything other than an instance of the appropriate class, the consistency checks will fail. When using a GUI such as Protégé, the input forms do not provide an option to select a “type” value for an instance of Variant other than from the appropriate class.
- Guess what? We forgot to give at least one value of Locator for our new Topics “Giacosa” and “Tosca”. We will generate an error indicating this when we run the consistency checking.

## § 9 OWL capabilities for TM authoring

### 9.1 Consistency Checks and Automatic Inferencing

As described throughout the examples above, by using a reasoner such as RACER with the Protege application, consistency checks and automatic inferencing to ensure TMDM compliance are available. These are activated simply by selecting the appropriate services from the Protege menu.

### 9.2 Additional Constraints

The TMDM ontology is set up in such a way to enable the user to easily add the kind of constraints described in the use cases of the TMCL requirements.

For instance: "Topic T can be used for Typing Occurrences, not for typing anything else".

Topic T should be declared as an instance of the "OccurrenceType" subclass. Occurrences are only allowed to get values for the "type" property from this Subclass, and no other Topic Map construct is allowed to get its "type" from this Subclass. To rule out the possibility that any Topic might be an instance of both SubClasses at the same time, the user can add an axiom to force the SubClasses within the "Type" subclass of "Topic" to be disjoint from each other.

"Occurrence of type O can only be used within scope S":

The user should create a subclass of Occurrence, which has "type" property set to an instance of OccurrenceType representing "O", and "hasScope" property mapped to an instance of Scope representing "S". Instances added to this subclass of Occurrence will automatically be of type "O" and have Scope "S". If the intention is that this is the only value of Scope these Occurrences may have, then the "hasScope" property for the created subclass should have its cardinality set to 1, to prevent it being linked to other Scopes via the "hasScope" property.

It should be mentioned that many of the TMCL Use Cases are related to "Topics of Type T.....": these are not discussed here due to the issue discussed in the Section 10 Outstanding Issues. It is worth noting though, that if the TMDM model were to be adapted to allow Topics to have types in the same way that Occurrences, Associations and so on have types, these cases would be able to be treated in a similar manner to the above examples.

Putting aside the above issue, the kinds of constraints required by the TMCL would not cause the authored Topic Map to go from OWL-DL to OWL-Full, although the user is not prevented from using the full available OWL buffet of constraint expressivity to add their own more creative constraints, which might cause their ontology to move into OWL-Full territory. To assist, Protege has a function which will tell the user which OWL-sublanguage the ontology is currently using, so that if one desires to stay within OWL-DL, one may check that this is the case when adding constraints. It is also likely that creative constraints would be at least partially lost when porting the Topic Map ontology to a Topic Map application based on a TMDM/TMCL platform, as there would be no way to translate these into TMCL.

### 9.3 Querying

Topic Maps authored using the TMDM ontology can be queried in Protege using a special Querying Plugin Tool which has a GUI frontend. Complex queries can be constructed by nesting simpler ones, and may make use of OR and AND logic through "Match Any" and "Match All" buttons provided.

The format simple queries in Protege is: [Class] [Slot] contains [Individual/Data value] The choices are presented from those that are validly available - that is if one chooses the [Class] to be "Topic", then the choices for [Slot] are only those properties where "Topic" is within the domain of the Property.

For instance for the Italian Opera Topic Map above, one can construct the following three simple queries:

1. Retrieve all instances of class [Topic] where slot [playRole] contains [Association Role X]
2. Retrieve all instances of class [AssociationRole] where slot [roleInAssociation] contains [Association Y]
3. Retrieve all instances of class [Association] where slot [type] contains [Association Type Y]

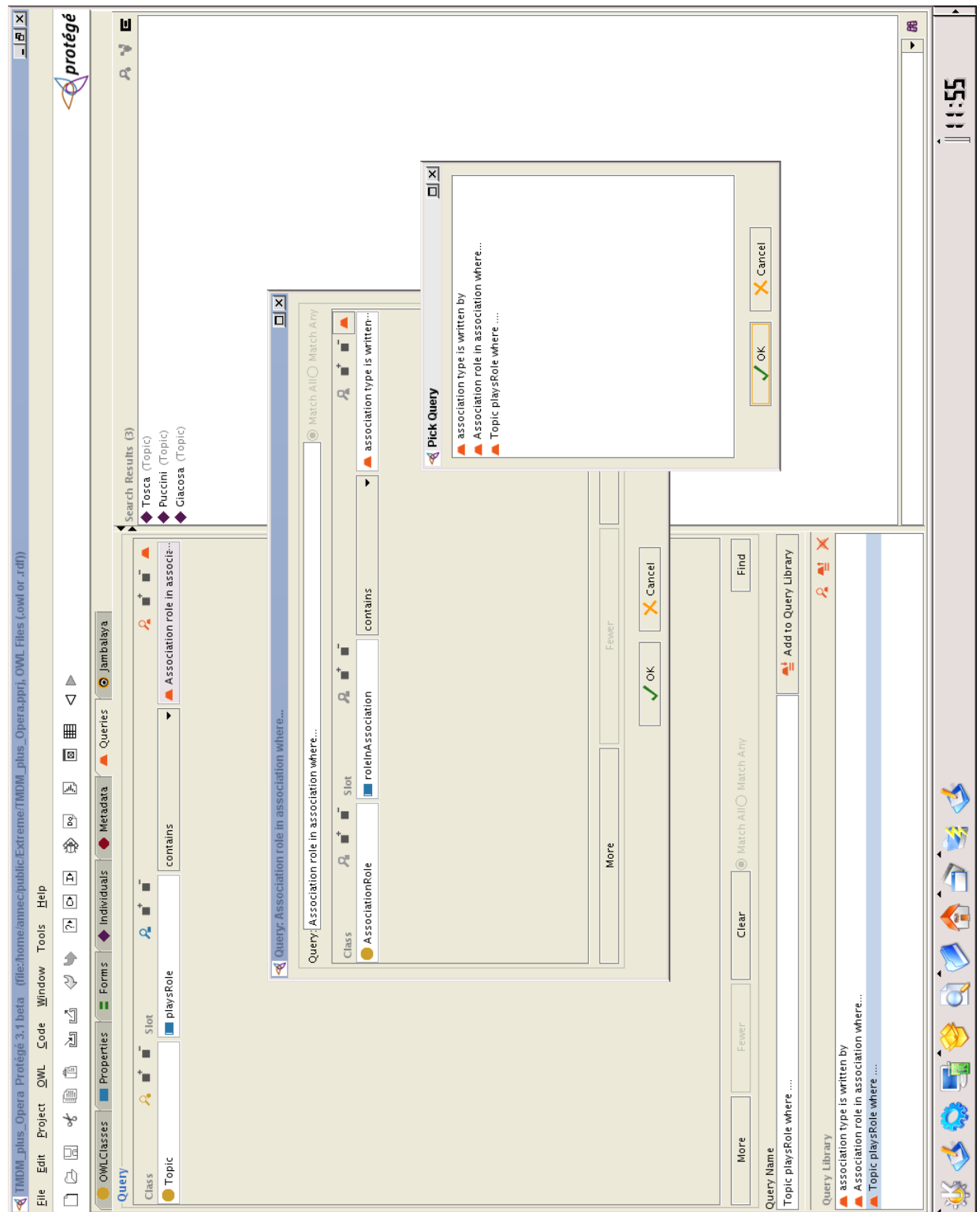
and then nest them, the third one being innermost.

By setting the value of AssociationType to "written\_by" in the innermost query, the query finds all Topics which play Roles in Associations of Type "written\_by". Based on the Italian Opera Topic Map ontology

described in the previous section, the result would be the Topics "Puccini," "Giacosa" and "Tosca" because they all play roles in the only Association of type "written by" (namely the Association "Puccini\_and\_Giacosa\_Wrote\_Tosca").

A query screenshot from this query being run in Protege is shown below:

Figure 4



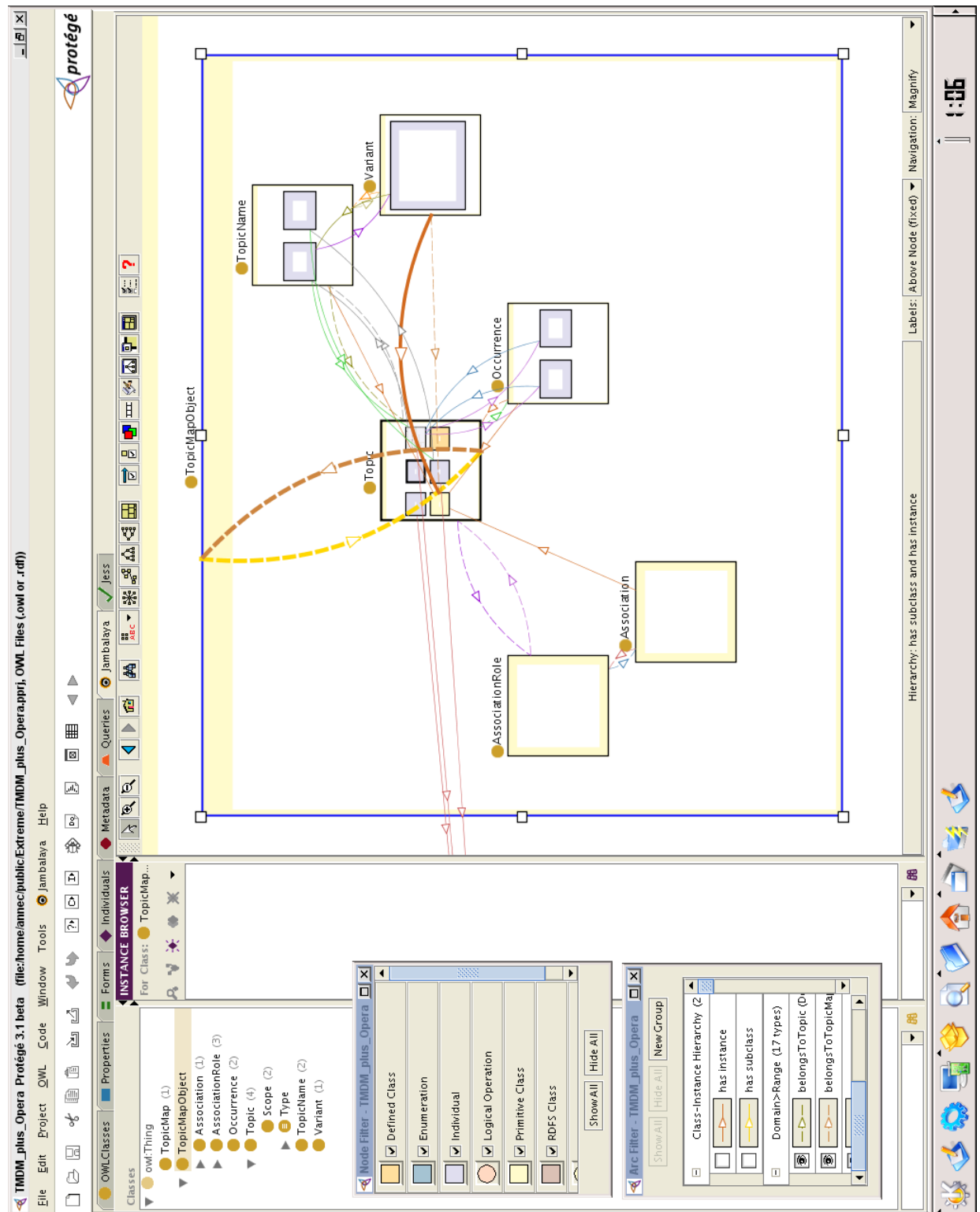


### 9.4 Visualisation

Protege provides several plugins for visualisation of ontologies, for instance Jambalaya. It provides various views of the ontology, and one may navigate through the Ontology by clicking on the Classes and Instances one wishes to view. The user may choose for views to be filtered to only contain certain constructs, and to be presented in different ways eg radially, as a tree structure, etc.

A sample screenshot produced by Protege's Jambalaya plugin is shown below, showing only one of the smorgasbord of possible views:

Figure 5



## § 10 Outstanding Issues

Up to this point we have implemented a solution which appears to successfully support Topic Map authors to build TMDM compliant Topic Maps as OWL ontologies. However, having put aside the issue of the functionality of "Type-Instance" and "SuperType-SubType" Associations as specified in the TMDM earlier, we must now return to see what our options are to get over the line in implementing ALL the required TMDM functionality.

Within the context of the approach presented in this paper, the choices to complete the functionality seem to break into three camps, which are described below.

### 10.1 The OWL-Full approach

In this approach, we do not implement Type-Instance and SuperType-SubType as instances of Associations, but instead use OWL's builtin `instanceOf` and `subclassOf` relationships to express these relations in a normal ontological manner. This requires the use of OWL-Full, since any topic may be treated as either a class or an instance in these relationships.

(Even if we could separate into classes and instances with no overlap, there is another reason for going beyond OWL-DL: we need to be able to assert characteristics of topics which are acting as classes without those characteristics being inherited by the topics which are their instances — for example, we want a topic's subject identifier to belong to it only and NOT to all the instances it is being used to type. However, there is no means to assert Data or Object Properties on classes, but only on the individuals contained within them. Thus any property asserted to on a class must be inherited by every individual in the class).

We dispense with the TMDM notion of building associations and roles with specific types to represent these relationships, and consider the "Type-Instance" and "SuperType-SubType" sufficiently grounded through being builtin OWL constructs. In theory this produces all the required functionality, however, in practice we now lack tools which support our ontology well enough to do all the consistency checking and querying we would like. However, better support for OWL-Full is on the W3C's agenda and if we revisit the issue in future we may find that the tools we need have become available.

We live with the fact that our Topic maps may be undecidable ie that sometimes our queries don't terminate or that not all consistency checking can give conclusive results. Since we are creating self-referential loops in this solution, this seems only fair.

### 10.2 The "Quick Fix" Workaround

As with the previous approach, we drop the notion of these relations being special Associations, but here Topics remain individuals at all times so that they can have all the Object and Data Properties they need. To support the Type behaviour, the user may create Class hierarchies within the Class "Topic", in the normal OWL manner, using OWL's `InstanceOf` and `SubClassOf` builtin properties. These classes may then simply be connected to the corresponding Topics which we want to have this Typing behaviour via a specially created ObjectProperty for this purpose. It is a little ugly and creates an extra step for the user in querying and constraint processing, but should mimic the desired functionality, and is within OWL-DL.

### 10.3 The Extra Java code approach

In this approach, we use a Protege plugin such as JESS to write extra Java code to produce the required functionality. For authoring this would then have to be downloaded by the TM author along with the TMDM ontology, and run in a similar way to the running of consistency checking and automatic inferencing (effectively it defines additional automatic inferencing). The functionality is produced by creating the flow-on relationships required, which are then available for subsequent functions such as querying and visualisation. Within this approach there are two paths, depending on how close we want to stay to the TMDM.

#### 10.3.1 Using Object Properties

For this path, as with the approaches above, we dispense with the TMDM notion of building associations and roles with specific types to represent these relationships. We define the "Type-Instance" and "SuperType-SubType" relations not by using OWL built-in constructs, but as special Object Properties

between Topics, in a similar way to the way Types are defined for other Topic Map objects in the previous sections. We remain in OWL-DL, as all Topics are always Individuals.

Taking this approach, we are able to use the built-in transitivity property within OWL to get the required behaviour for the "SuperType-SubType" Object Property, and just need a little extra code to ensure that Instances of Types get passed upwards to their SuperTypes.

### 10.3.2 Generating Instances of Association and Association Roles

On this path, we stay with the letter of the TMDM specification and express "Type-Instance" and "SuperType-SubType" relations as special instances of associations as the TMDM states, then use extra Java code as above to exhibit the required behaviour. This means creating extra instances of Associations and Association Roles on the fly to accommodate the flow-on relations.

The TMDM ontology as it has been described in previous sections can easily accommodate the representation through the use of subClasses in combination with constraints, and the definition of special instances of Topics for the types. Constraints are used to ensure there are exactly two Roles of two particular Types in each instance of an Association which has its type set to one of these special Association Types. Special subclasses are created to ensure the Subject Identifiers for the Types of the Roles and Associations are set appropriately. The full construction in Protege is available at the authors website.

The additional Java Code is then required to produce the extra instances of Associations and Association Roles which make the SuperClass-Subclass Association transitive, and to ensure that Instances of Types in "Type-Instance" Associations get passed upwards to any Topics playing the roles of "SuperType" in instances of "SuperType-Subtype" Associations where where the original Topic is playing the role of "SubType".

## § 11 Conclusions

This paper presents a proposal for an OWL-DL formalisation of the ISO drafted standard Topic Map Data Model. It illustrates the fit between OWL and the TMDM, and the potential for the use of OWL-DL as a medium for representing the TMDM constructs, providing the basis for end-users to author Topic Maps in OWL-DL. The use of OWL-DL and its associated tools for constructing Topic Maps provides significant advantages over previous Topic Map representations in terms of explicit specification, formal semantics, constraint checking and querying capabilities. An outstanding issue relating to the implementation of Type-Instance and SuperType-SubType relations as specified in the TMDM was identified, and some approaches for implementing the required functionality were outlined.

## § 12 Appendix A

TMDM Ontology: OWL-DL code generated by Protege:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about="" />
  <owl:Class rdf:ID="AssociationRole">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:allValuesFrom>
          <owl:Class rdf:ID="AssociationRoleType" />
        </owl:allValuesFrom>
        <owl:onProperty>
          <owl:FunctionalProperty rdf:ID="type" />
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
          1</owl:cardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="playedBy" />
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
```

```

    <owl:ObjectProperty rdf:ID="roleInAssociation"/>
  </owl:onProperty>
  <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</owl:cardinality>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Class rdf:ID="TopicMapObject"/>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Scope">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Topic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#AssociationRoleType">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Type"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Topic">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="belongsToTopicMap"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:minCardinality>
    <owl:onProperty>
      <owl:DatatypeProperty rdf:ID="locator"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#TopicMapObject"/>
</owl:Class>
<owl:Class rdf:ID="TopicName">
  <rdfs:subClassOf rdf:resource="#TopicMapObject"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom>
        <owl:Class rdf:ID="TopicNameType"/>
      </owl:allValuesFrom>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="#type"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="belongsToTopic"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:allValuesFrom rdf:resource="#Scope"/>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="hasScope"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="TopicMap">
<owl:Class rdf:ID="OccurrenceType">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Type"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="AssociationType">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Type"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Variant">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasScope"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="#Scope"/>
    </owl:Restriction>

```

```

</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:minCardinality>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="variantScope"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#TopicMapObject"/>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="isVariantOf"/>
    </owl:onProperty>
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Association">
  <rdfs:subClassOf rdf:resource="#TopicMapObject"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#belongsToTopicMap"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="#type"/>
      </owl:onProperty>
      <owl:allValuesFrom rdf:resource="#AssociationType"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom rdf:resource="#Scope"/>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasScope"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasRole"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Occurrence">
  <rdfs:subClassOf rdf:resource="#TopicMapObject"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="occurrenceOfTopic"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom rdf:resource="#Scope"/>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasScope"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Type">
  <owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#AssociationType"/>
        <owl:Class rdf:about="#AssociationRoleType"/>
        <owl:Class rdf:about="#OccurrenceType"/>
        <owl:Class rdf:about="#TopicNameType"/>
      </owl:unionOf>
    </owl:Class>
  </owl:equivalentClass>
</rdfs:subClassOf rdf:resource="#Topic"/>

```

```

</owl:Class>
<owl:Class rdf:about="#TopicNameType">
  <rdfs:subClassOf rdf:resource="#Type"/>
</owl:Class>
<owl:ObjectProperty rdf:about="#belongsToTopicMap">
  <rdfs:range rdf:resource="#TopicMap"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Topic"/>
        <owl:Class rdf:about="#Association"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasTopicName">
  <rdfs:domain rdf:resource="#Topic"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:about="#belongsToTopic"/>
  </owl:inverseOf>
  <rdfs:range rdf:resource="#TopicName"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#playedBy">
  <rdfs:domain rdf:resource="#AssociationRole"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:ID="playsRole"/>
  </owl:inverseOf>
  <rdfs:range rdf:resource="#Topic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#roleInAssociation">
  <owl:inverseOf>
    <owl:ObjectProperty rdf:about="#hasRole"/>
  </owl:inverseOf>
  <rdfs:domain rdf:resource="#AssociationRole"/>
  <rdfs:range rdf:resource="#Association"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#occurrenceOfTopic">
  <rdfs:range rdf:resource="#Topic"/>
  <rdfs:domain rdf:resource="#Occurrence"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:ID="hasOccurrence"/>
  </owl:inverseOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasOccurrence">
  <owl:inverseOf rdf:resource="#occurrenceOfTopic"/>
  <rdfs:range rdf:resource="#Occurrence"/>
  <rdfs:domain rdf:resource="#Topic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#variantScope">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasScope"/>
  </rdfs:subPropertyOf>
  <rdfs:range rdf:resource="#Scope"/>
  <rdfs:domain rdf:resource="#Variant"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#playsRole">
  <rdfs:domain rdf:resource="#Topic"/>
  <owl:inverseOf rdf:resource="#playedBy"/>
  <rdfs:range rdf:resource="#AssociationRole"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasVariant">
  <rdfs:range rdf:resource="#Variant"/>
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasScope"/>
  </rdfs:subPropertyOf>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:about="#isVariantOf"/>
  </owl:inverseOf>
  <rdfs:domain rdf:resource="#TopicName"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasScope">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#TopicName"/>
        <owl:Class rdf:about="#Variant"/>
        <owl:Class rdf:about="#Occurrence"/>
        <owl:Class rdf:about="#Association"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Scope"/>
        <owl:Class rdf:about="#TopicName"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
  <owl:inverseOf>

```

```

    <owl:ObjectProperty rdf:ID="isScopeOf"/>
  </owl:inverseOf>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasRole">
  <rdfs:domain rdf:resource="#Association"/>
  <rdfs:range rdf:resource="#AssociationRole"/>
  <owl:inverseOf rdf:resource="#roleInAssociation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#isScopeOf">
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Occurrence"/>
        <owl:Class rdf:about="#Association"/>
        <owl:Class rdf:about="#TopicName"/>
        <owl:Class rdf:about="#Variant"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
  <owl:inverseOf rdf:resource="#hasScope"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Scope"/>
        <owl:Class rdf:about="#TopicName"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#isVariantOf">
  <rdfs:domain rdf:resource="#Variant"/>
  <owl:inverseOf rdf:resource="#hasVariant"/>
  <rdfs:range rdf:resource="#TopicName"/>
  <rdfs:subPropertyOf rdf:resource="#isScopeOf"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#belongsToTopic">
  <rdfs:domain rdf:resource="#TopicName"/>
  <owl:inverseOf rdf:resource="#hasTopicName"/>
  <rdfs:range rdf:resource="#Topic"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="subjectLocator">
  <rdfs:domain rdf:resource="#Topic"/>
  <rdfs:subPropertyOf>
    <owl:DatatypeProperty rdf:about="#locator"/>
  </rdfs:subPropertyOf>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#anyURI"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="sourceLocator">
  <rdfs:subPropertyOf>
    <owl:DatatypeProperty rdf:about="#locator"/>
  </rdfs:subPropertyOf>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#anyURI"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="variantName">
  <rdfs:domain rdf:resource="#Variant"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#locator">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#anyURI"/>
  <rdfs:domain rdf:resource="#TopicMapObject"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="occurrenceTextLocatedAt">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#anyURI"/>
  <rdfs:subPropertyOf>
    <owl:FunctionalProperty rdf:ID="occurrenceText"/>
  </rdfs:subPropertyOf>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="subjectIdentifier">
  <rdfs:subPropertyOf rdf:resource="#locator"/>
  <rdfs:domain rdf:resource="#Topic"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#anyURI"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="baseLocator">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#TopicMap"/>
</owl:DatatypeProperty>
<owl:FunctionalProperty rdf:ID="variantNameLocatedAt">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#anyURI"/>
  <rdfs:subPropertyOf rdf:resource="#variantName"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="variantNameString">
  <rdfs:subPropertyOf rdf:resource="#variantName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="topicNameString">

```



```

<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:domain rdf:resource="#TopicName"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="occurrenceTextString">
  <rdfs:subPropertyOf>
    <owl:FunctionalProperty rdf:about="#occurrenceText"/>
  </rdfs:subPropertyOf>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#occurrenceText">
  <rdfs:domain rdf:resource="#Occurrence"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#type">
  <rdfs:range rdf:resource="#Type"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
<rdfs:domain>
  <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#TopicName"/>
      <owl:Class rdf:about="#AssociationRole"/>
      <owl:Class rdf:about="#Occurrence"/>
      <owl:Class rdf:about="#Association"/>
    </owl:unionOf>
  </owl:Class>
</rdfs:domain>
</owl:FunctionalProperty>
<owl:InverseFunctionalProperty rdf:ID="reifiedBy">
  <rdfs:range rdf:resource="#Topic"/>
  <owl:inverseOf>
    <owl:InverseFunctionalProperty rdf:ID="reifierOf"/>
  </owl:inverseOf>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#TopicMapObject"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="#reifierOf">
  <rdfs:domain rdf:resource="#Topic"/>
  <rdfs:range rdf:resource="#TopicMapObject"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <owl:inverseOf rdf:resource="#reifiedBy"/>
</owl:InverseFunctionalProperty>
</rdf:RDF>

```

---

## Bibliography

- [OWL] Mike Dean, Guus Schreiber, editors, *OWL Web Ontology Language Reference*, W3C Recommendation, 10 February 2004 <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>
- [Protege] Stanford University *The Protege Ontology Editor and Knowledge Acquisition System*, Open Source Ontology Editor [protege.stanford.edu/](http://protege.stanford.edu/)
- [RACER] Volker Haarslev, Ralf Moller, Michael Wessel, *RACER*, Semantic Middleware for Industrial Projects based on RDF/OWL <http://www.sts.tu-harburg.de/~r.f.moeller/racer/>
- [RDF] Dave Beckett, editor, *RDF/XML Syntax Specification (Revised)*, W3C Recommendation, 10 February 2004 <http://www.w3.org/TR/rdf-syntax-grammar/>
- [RDFTM] Steve Pepper, Fabio Vitali, editors, *RDFTM: Survey of Interoperability Proposals*, W3C RDF/Topic Maps Inter-operability Task Force publication, Editors Draft, 24 February, 2005 <http://www.w3.org/2001/sw/BestPractices/RDFTM/survey-2005-02-24>
- [TM] M. Biezunski, M. Bryan, S. Newcomb, editors, *ISO/IEC 13250, Topic Maps (Second Edition)* ISO 13250 International Standard, 22 May 2002 <http://www.y12.doe.gov/sgml/sc34/document/0322.htm/>
- [TMCL] Graham Moore, Mary Nishikawa, Dmitry Bogachev, *Topic Map Constraint Language (TMCL) Requirements and Use Cases* ISO 13250 Editors Draft, 16 October, 2004 <http://www.jtc1sc34.org/repository/0548.htm>
- [TMDM] Lars Marius Garshol, Graham Moore, *Topic Maps - Data Model* ISO 13250 Final Committee Draft, 10 January, 2005 <http://www.isotopicmaps.org/sam/sam-model/>
- [TMQL] Lars Marius Garshol, Robert Barta, *TMQL Requirements* ISO 13250 Draft, 7 November, 2003 <http://www.y12.doe.gov/sgml/sc34/document/0448.htm>

[TMRM] Patrick Durusau, Steve Newcomb, editors *Topic Maps - Reference Model* ISO 13250 Committee Draft, 12 February, 2005 <http://www.isotopicmaps.org/TMRM/>

[XTM1.0] Steve Pepper, Graham Moore, editors *XML Topic Maps (XTM) 1.0* TopicMaps.Org Specification, 6 August, 2001 <http://www.topicmaps.org/xtm/1.0/>

---

## The Author

### Anne Cregan

*University of New South Wales, Knowledge Representation and Reasoning Program,  
National Information and Communications Technology Australia (NICTA) Centre of  
Excellence and Artificial Intelligence Laboratory, School of Computer Science and  
Engineering*

Sydney

NSW

Australia

2052

[annec@cse.unsw.edu.au](mailto:annec@cse.unsw.edu.au)

<http://www.cse.unsw.edu.au/~annec/>

Anne Cregan is a PhD student in NICTA's Knowledge Representation and Reasoning Program. Prior to PhD enrolment she worked as a Technology and Business Consultant, Data Miner and Marketing Modeller. She was General Manager of two Internet-related companies in 1998-2000. She has a first class honours degree from the University of Sydney in Cognitive Psychology and Human Intelligence.

### Extreme Markup Languages 2005®

Montréal, Québec, August 1-5, 2005

*This paper was formatted from XML source via XSL  
by Mulberry Technologies, Inc.*