Managing Literature References with Topic Maps

Robert Barta
IT School
Bond University
rho@bond.edu.au

Kate Kelly
Library Services
Bond University
kkelly@bond.edu.au

Abstract

This article introduces Topic Map (TM) authoring and ontology engineering. With a running example of a simple TM-based literature reference database we show how a localized ontology can be defined to constrain and describe our knowledge domain. We then use the same technique to describe the structure of the BibTeX *format. These two ontologies can then be used to formalize a mapping between them. For this purpose we use an experimental TM query language.*

Keywords Topic Maps, References, Ontology

1 Introduction

In the context of literature references the Topic Map standard[2] can be used to store meta data along with references to external objects. One of the objectives was also that users can have highly localized concepts but also can use globally defined ones to simplify exchange and conciliation of multiple maps.

In the following we elaborate on how TM concepts and technologies can be used to provide an open and manageable framework to define and operate with ontologies.

2 TM authoring

Topic Maps are a relatively new technology in the Semantic Web arena[13]. Their most obvious difference to RDF[12] is that they use a two-level approach[9, 10]: The more lexicographical part of a topic map consists of topics which represent (reify) real world objects but also abstract concepts. Here the main focus is on naming issues for different contexts and also URIs to objects external to the map.

The semantic aspect of TMs is covered with associations. Other than RDF statements they are not (subject, predicate, object) triples, but coerce any number of topics together, whereby each of these topics plays a specific role.

Proceedings of the 7th Australasian Document Computing Symposium,
Sydney, Australia, December 16, 2002.

2.1 Topics

Using the AsTMa=[3] notation for compactness, the following TM fragment defines a topic:

Listing 1: Topic definitions

```
ltm-spec (l-specification) reifies \
   http://www.ontopia.net/.../ltm.html
bn: LTM, The Linear Topic Map Notation
bn @ latex: {LTM}, The Linear Topic \
   Map Notation
oc (cite-code) @latex: urn:bibtex:lmg01
in: This technical report defines ...
```

It carries the id `ltm-spec` and is an instance of the concept `l-specification`. It has a base name (indicated by `bn`) reifying the online document at the given URL.

Other information a topic may contain is inline data (`in`) which may contain descriptive text, and occurrences (`oc`) which hold URIs. Inline data and occurrences can be typed, as it is the case with the `cite-code` occurrence above.

2.2 Associations

Associations are statements about the relationships of various topics. The following

```
(is-author-of)
opus   : ltm-spec
author : p-lars-marius-garshol
```

would state that "the topic `p-lars-marius-garshol` plays the role of the author in an `is-author-of` association whereby `p-lars-marius-garshol` plays the role of an `opus`". All topics have been referenced via an URI and are declared separately.

3 TM Ontology Engineering

Ontologies—once defined—can be used to provide formal and informal rules how to create TM documents. In an integrated TM authoring environment ontologies can be used to guide the authoring process in the same way as XML schemas are used for XML authoring. Other uses include filtering

of TM documents according to their ontology conformance or reconciliation of heterogeneous data sources[7]. As in the following we can also use ontologies to create a *projection*, i.e. a particular view into a topic map.

3.1 Source Ontology

While there exists a general set of requirements for an ontology language[8], at the time there are only proposals for TM ontology specification languages (e.g. [11]). In the following we use AsTMa![5] which extends the authoring language AsTMa= with regular expressions, quantifiers and boolean operators.

To define an ontology $\mathcal{L}$ for the literature references we first have to set up some basic vocabulary and taxonomy:

```
(is-subclass-of)
subclass: l-book
superclass: l-document
# similar for articles, reports, ...
```

Aside from the vocabulary we have to set up rules on individual literature references. The rules can either prescribe (MUST), suggest (MAY) or forbid (MUST NOT) particular patterns found in a conforming topic map.

In our case we police that every document should have an author:

```
every [ $t (l-document)* ]
  => exists ] (is-author-of)
              opus   : $t
              author : $a [
```

First we single out all topics which are instances of `l-document`. The `*` symbolizes that these can be direct instances or instances of a subclass of `l-document`. For all those documents we prescribe the existence of an appropriate association. The][around the association pattern has to be read as *exactly so*, while the [] allow a more liberal interpretation.

Using this technique we add more rules to build the complete ontology.

3.2 Target Ontology

In this section we discuss one particular application where we use our literature ontology to filter only specific aspects out of a topic map. In this process we have to define a mapping between literature reference information in Topic Map form and a conventional database system which follows the entity-attribute paradigm.

Our database format will be BibTeX[6] which prescribes a set of document classes (books, reports, ...). Each of these classes consists of specific mandatory or optional attributes (author, title, ...).

The topic `ltm-spec` defined in listing 1 would be represented in BibTeX as follows:

```
@misc{urn:bibtex:lmg01,
      author = {{Garshol, Lars Marius}},
      title = {{LTM}, The Linear Topic
               Map Notation},
      year = {2001},
      url = {http://www.ontopia.net/...}
}
```

As such, BibTeX follows its own schema definition which serves there the role of an ontology. To allow for a formalized mapping within one single formalism, we have captured this BibTeX schema in an AsTMa! constraint by characterizing the document classes and its specific attributes.

```
b-book (class)

b-report (class)

publisher (attribute)
```

Given that, we can now make explicit the rules specific for BibTeX, such as that a book must have a title (we conveniently store that in a base name):

```
every [ $b (book) ] => exist [ $b
                                 bn: * ]
```

All other attributes we plan to save via an generic `is-attribute-of` association. We only have to define that all these associations must have a particular layout:

```
every $a [ (is-attribute-of) ]
   => exists $a ] (is-attribute-of)
                  object    : *
                  attribute : *
                  value     : * [
```

A more application specific rule would be that every book also must have a `publisher` attribute while we do not care about its value:

```
every [ $b (book) ]
  => exists [ (is-attribute-of)
              object    : $b
              attribute : publisher ]
```

In a similar way we proceed with all other attributes and all other BibTeX classes.

4 Ontology Mapping

To mediate between $\mathcal{L}$ and the BibTeX ontology $\mathcal{B}$ we can hardcode the mapping directly into an application. This was actually done, not only to get working code but also to understand the practicalities involved.

4.1 Specification

In a first step the relevant topics of the literature ontology have to be identified as those which are a direct or indirect subclasses of `l-document`. For all these (`l-book`, `l-article`...) we have to define their respective counterparts in $\mathcal{B}$. While obviously a `l-book` in $\mathcal{L}$ will correspond with `b-book` in $\mathcal{B}$, for other document types in $\mathcal{L}$ this choice is less obvious.

According to $\mathcal{B}$ the class then will define which attributes are mandatory and which are optional for this object. For the book `title`, `publisher`, `year` and `author` or `editor` have to be defined, whereas the `volume` and other attributes are optional.

For all the above attributes values have to be identified in the source map. For the example specification document provided in listing 1, the application will have to follow all `is-author-of` associations for that particular document to identify the topics playing the author role there. The base names of the respective topics will be used as value for the author.

One complication is due to the fact that BibTeX citations need a *cite code* which we suggest to exist in $\mathcal{L}$:

```
every [ $t (document)* ]
  => suggested exists
       [ $t
         oc @ latex (cite-code): * ]
```

As the code is only useful in a particular context, we have added a scope `latex` to restrict its validity to that scope. We strengthen this code to be unique within one map:

```
every [ $t1
         oc (cite-code): $code ]
  => not exists
       [ $t2
         oc (cite-code): $code ]
```

This rule first singles out all topics having a cite code; they will be bound to the variable `$t1` and the corresponding value of the code is bound to `$code`. In the second clause it will then be checked whether there is a topic which contains an identical cite code. The—somewhat unorthodox—AsTMa! semantics enforces that two differently named variables cannot be bound to the same values, so the topic ids must be different.

Another issue involves BibTeX expecting a particular layout style for some attributes. Titles, for instance, have to follow a particular capitalization depending on the document class.

As it is difficult to formalize these rules in $\mathcal{B}$, we will have to burden the author of a map conforming to $\mathcal{L}$ to provide appropriate input:

```
every [ $t (document)* ]
  => suggested exists
       [ $t
         bn @ latex : * ]

every [ (is-author-of)
         author: $a ]
  => suggested exists
       [ $a
         bn @ latex : * ]
```

If the mapping application finds a LaTeXvariant of the title, then that should have preference over an unscoped title. In the similar way author naming can be tailored for LaTeX.

4.2 Formalization

With these specifications above a dedicated application can now perform the mapping. One of the promises of a uniform formalism, though, is that such a mapping between two ontologies can be defined *within* that formalism. For this purpose we make use of an experimental TM query language, AsTMa?[4].

In the same way as SQL operates on tables to return a table and XQuery[1] operates on XML documents to return an XML document, AsTMa? queries analyze topic maps following the source ontology and return maps conforming to the target ontology.

As an introductory example, let us consider the conversion of books together with their titles:

```
in "literature.tm"
  where
    exists [ $b (l-book)
               bn @ latex : $t ]
  return
    {$b} (b-book)
    bn: {$t}
```

Here we iterate over the sourced topic map and look for all submaps which conform to the condition provided by the *where* clause. We are selecting all submaps which contain a `l-book` topic with a LaTeX-ready title. According to the AsTMa? language semantics only those submaps will be considered which are minimal in that they do not contain unnecessarily other topics or associations while still be conforming. In our case the submaps will only consist of the `l-book` topics. Any duplicates will then be discarded.

For all these submaps the *return* clause is evaluated. The return clause contains AsTMa= code, this time for constructing a new map. We reuse the topic id of a particular `l-book` topic also as id for topic in the target map. As all these ids are bound to the variable `$b`, they can be referred to as {`$b`}.

A more sophisticated query would capture more information about a book:

```
in "literature.tm"
  where
    exists [ $b (l-book)
             bn @ latex : $t ]
       and
    exists [ (is-author-of)
             opus   : $b
             author : $p

             $p (person)
             bn @ latex : $n ]*
  return
    {$b} (b-book)
    bn: {$t}

    (is-attribute-of)
    class     : {$b}
    attribute : author
    value     : {join(", ", $n)}
```

Again we use first a pattern to identify one topic being an `l-book`. In the second *exists* clause we now identify the association which links the author to the book `$b`. A subtle difference is the symbol `*` trailing this pattern. With this we signal *greedy matching* to the TM processor, i.e. that the matched submap should have as many instances of this pattern as possible. This results then in a list of matches. As before the processor will only pass through those minimal maps which do not violate the *where* clause (no junk).

In the construction part within the *return* clause we again refer to a single book adding the matched title as base name. In accordance with $\mathcal{B}$ we also generate an association `is-attribute-of` to add the author information which we have matched before. As we have matched multiple names the processor will have captured the individual names within a list `$n`. We concatenate these strings in the list and use the result as attribute value.

Once the target map has been built it is trivial to convert this into the final BibTeX text format.

5 Conclusion

We have demonstrated how TM engineering can be used to manage structured content and how ontologies can be used to constrain the content. The declarative nature of ontologies allows us to freely combine ontologies. Thus a document which satifies two ontologies can be said to satisfy a combination of the two.

Then we formalized a simple entity-attribute model into an ontology using generic association. This was the basis to formalize the mapping between the two ontologies which would enable a generic query processor to translate a topic map from one ontology into another.

The references for this article are mastered via the aforementioned prototype.

References

[1] XQuery, W3C Working Draft 16 august 2002. W3C.

[2] XML Topic Maps (XTM) 1.0 Specification. TopicMaps.Org, 2001.

[3] Barta, R. AsTMa= language definition, technical report. Bond University, 2001. http://www.it.bond.edu.au/publications/02TR/02-14.pdf.

[4] Barta, R. AsTMa? (Asymptotic Topic Map Notation, Querying), tutorial. Bond University, 2002. http://astma.it.bond.edu.au/astma%3F.dbk.

[5] Barta, R. AsTMa! language definition, technical report. Bond University, 2002. http://astma.it.bond.edu.au/astma!-spec.dbk.

[6] H. Kopka, P. W. Daly. A Guide to LaTeX. *Addison Wesley*, 1999.

[7] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Huebner. Ontology-based integration of information - a survey of existing approaches. *Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing', Seattle, WA, pages 108-117.*

[8] J. Heflin, R. Volz, J. Dale. Requirements for a Web ontology language, W3C working draft 08 july 2002.

[9] Jonathan Robie. The syntactic web - syntax and semantics on the web. *XML 2001*, 2001.

[10] Lacher,-M.-S.; Decker,-S. Rdf, topic maps, and the semantic web. *Markup-Languages:-Theory-&-Practice. Summer 2001; 3(3): 313-31*, 2001.

[11] Lars M. Garshol. The Ontopia Schema Language. ISO/IEC JTC 1/SC34, Information Technology – Document Description and Processing Languages.

[12] O. Lassila and K. Swick. Resource description frame-work (RDF) model and syntax specification, technical report, W3C. 1999.

[13] T. Berners-Lee. Feature article: The semantic web. *Scientific American*, 2001.