Methods for Porting Resources to the Semantic Web

Bob Wielinga[1], Jan Wielemaker[1], Guus Schreiber[2], and Mark van Assem[2]

[1] University of Amsterdam
Social Science Informatics (SWI)
Roetersstraat 15, 1018 WB Amsterdam, The Netherlands
{wielinga,jan}@swi.psy.uva.nl
[2] Vrije Universiteit Amsterdam
Department of Computer Science
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{schreiber,mark}@cs.vu.nl

Abstract. Ontologies will play a central role in the development of the Semantic Web. It is unrealistic to assume that such ontologies will be developed from scratch. Rather, we assume that existing resources such as thesauri and lexical data bases will be reused in the development of ontologies for the Semantic Web. In this paper we describe a method for converting existing source material to a representation that is compatible with Semantic Web languages such as RDF(S) and OWL. The method is illustrated with three case studies: converting Wordnet, AAT and MeSH to RDF(S) and OWL.

1 Introduction

Semantic Web applications will require multiple large ontologies for indexing and querying [5]. Developing such ontologies is a time consuming and costly process, so we assume that in general these ontologies will not be developed from scratch. Rather, existing resources such as thesauri, lexical data bases or ontologies published in a proprietary format will be used as sources for development of ontologies for the Semantic Web. In this paper we describe a method for converting existing source material to a representation that is compatible with semantic web languages such as RDF(S) and OWL.

The problem that we address in this paper is: how can existing resources be converted to representations that can be understood by Semantic Web applications without altering the original material, and at the same time assign semantics to these representations that is (presumed to be) compatible with the intended semantics of the source. An important corrolary of this problem statement is that the transformation process from source material to Semantic Web ontology is transparant and traceable. Users of ontologies created through conversion processes will need to be aware of the interpretative steps that have taken place in the transformation process and may want to influence that process according to their own insights and requirements. So, although the conversion

of a single source to Semantic Web standards may not be a very difficult task, the underlying principles and methods are of great importance to the Semantic Web enterprise.

This paper is organised as follows. In Sect. 2 we describe the general requirements and methods for converting existing materials. Section 3 to Section 5 discuss three case studies that demonstrate various applications of the method.

2 General Method

The method for converting source material to ontologies is based on the general principle of fully automatic transformation of the source material in a number of steps. The first step (step 1a) in the conversion process is a structure-preserving syntactic translation from the source format to RDF(S). We assume that a data model of some sort is available of the source. This can be a conceptual model described in textual form, a template record structure, an XML DTD or a proper data model for example represented in UML. From the data model an RDF(S) schema is derived, where classes with properties are defined. It is recommended that naming conventions are preserved, with an exception for abbreviations which should be expanded. For example, the abbreviation "BT" for broader term, used in many thesauri, should be mapped to an RDF(S) property "broaderTerm". When the source is represented in XML some elements do not have to be represented as classes when they only serve as placeholders. For example the element "TermList" used in MeSH (see Section 5), can be directly mapped to the property "hasTerm" since RDF(S) properties can have multiple values.

Two complications may arise in the creation of the RDF schema. A first problem may occur when an XML DTD enforces a strict sequence through comma-separated element definitions. Only when the order is interpreted to be relevant the RDF list construct should be used, which can make the RDF representation somewhat complicated, since the representation of ordered relations as RDF lists requires special interpretation machinery. In general this should be avoided where possible. Although for example, the MeSH DTD states that a DescriptorRecord always has its children elements in strict order (comma), this is probably not required. Therefore, it is possible to translate DescriptorRecords by translating each child element and linking them to the Descriptor using properties.

A second complication may occur when data elements have internal substructures. For example, many dictionaries give multiple meanings under one headword, usually indicated by number codes. In such cases it has to be decided whether each subentry should be mapped onto a separate class or whether the subentries can be mapped to properties.

When an RDF(S) schema is established the data elements from the source can be translated to instances of the schema. In this structural translation step no information is lost or added, it concerns just a translation between the original format and RDF(S).

The next step (step 1b) in the conversion process concerns the *explication* of information that is implicit in the original data format but that is intended by the conceptual model. Examples of cases where explication can be applied are given below.

- Thesauri with an origin in the bibliographic sciences are often structured as a set of records, with fields for hierarchical relations. An example is MeSH, which has Descriptors with TreeNumbers. These TreeNumbers can be used to create (and are intended to signify) a hierarchy, e.g. by adding a subTree relation between Descriptors.
- Attributes in records often have terms as value, rather than unique identifiers. These terms have to be mapped to an xml namespace with a unique identifier.
- Some entries can play a special role. For example in AAT, some terms are "GuideTerms" that function as a structuring device in the hierarchy, but which are not supposed to be used for indexing. In AAT these terms are identified by enclosing them in brackets (<>). The special role of such entries can be made explicit by defining them as instances of a special class in the data model (e.g. "GuideTerm" as a subclass of "Term"). In this way the original intention of the conceptual model is preserved and made explicit.

The first two translation steps together form a syntactic conversion stage. A second stage in the conversion process concerns a semantic transformation. In the first step of the semantic conversion (step 2a) the RDF(S) instances generated in the syntactic stage are augmented according to the intended semantics of the source conceptual model. Many thesauri and lexical data bases intend their entries to be interpreted as a hierarchy of concepts. If the properties "broaderTerm" and "narrowerTerm" are used to represent the hierarchical relation they can be defined in OWL as inverse property of each other and as transitive properties.

In the second step (step 2b)of the semantic conversion the instances of the first stage are reinterpreted in terms of the RDFS or OWL semantics. For example the hierarchical relations of the thesaurus can be interpreted as RDF(S) "subClassOf" relations. This step adds semantics to the ontology (such as inheritance) which may or maynot have been intended by the creators of the source. A standard way to achieve this reinterpretation is to make the classes in the syntactic schema subclasses of class `Class` (i.e. meta classes) and making the hierarchical relations such as "subtreeOf" in Mesh and "broaderTerm" in other thesauri, a subproperty of "subClassOf". This creates an interpretation of the source as a proper subclass hierarchy. Other properties can also be mapped onto RDF(S) and OWL properties. For example a property such as "relatedTerm" which is present in many thesauri can be mapped onto "seeAlso" in RDFS/OWL.

Figure 1 summarizes the steps described above.

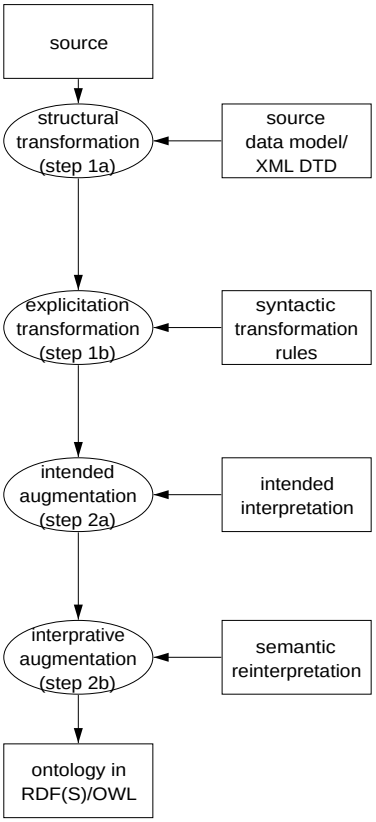

Fig. 1. Schematic representation of the conversion steps

3 Case 1: Wordnet

WordNet [1] is a large lexical data base, originally developed for linguistic purposes, but now an important resource for research on the Semantic Web. Step 1a of the conversion of WordNet to an RDF representation was performed by Decker and Melnik[3]. Their RDF Schema for WordNet defines classes and properties for the data model of WordNet. This means that WordNet *synsets* (the basic WordNet concepts) are represented as instances of the class `LexicalConcept` and that the WordNet hyponym relations (the subclass relations in WordNet) are represented as tuples of the `hyponymOf` relation between instances of `wns:LexicalConcept`. The data model and source representation of WordNet is quite explicit and clean, so step 1b is not required in this case. In step 2a additional properties of the representation could be defined. For example, the WordNet relation `wn:similarTo` could be asserted to be a subproperty of the OWL `owl:SymmetricProperty`. In our present implementation this step has not been performed.

The RDF(S) representation leads to a representational mismatch, as we are unable to treat WordNet concepts as classes and WordNet hyponym relations as subclass relations. This problem can be resolved by performing step 2b of the conversion process using RDFS metamodelling primitives. Consider the following two RDFS descriptions:

```
<rdf:Description rdf:about="&wns;LexicalConcept">
  <rdfs:subClassOf rdf:resource="&rdfs;Class"/>
</rdf:Description>

<rdf:Description rdf:about="&wns;hyponymOf">
  <rdfs:subPropertyOf rdf:resource="&rdfs;subClassOf"/>
</rdf:Description>
```

The first statement specifies that the class `LexicalConcept` is a subclass of the built-in RDFS metaclass `Class`, the instances of which are classes. This means that now all instances of `LexicalConcept` are also classes. In a similar vein, the second statement defines that the WordNet property `hyponymOf` is a subproperty of the RDFS `subClassOf` property. This enables us to interpret the instances of `hyponymOf` as subclass links.

We expect representational mismatches to occur frequently in any realistic Semantic Web setting. RDF(S) mechanisms similar to the ones above can be employed to handle this. However, this poses the requirement on the toolkit that the infrastructure is able to interpret subtypes of `rdfs:Class` and `rdfs:subPropertyOf`. In particular the latter was important for our applications, e.g., to be able to reason with WordNet hyponym relations as subclass relations and to visualize WordNet as a class hierarchy.

[3] http://www.semanticweb.org/library/

4 Case 2: AAT

The Art and Architecture Thesaurus (AAT[4] [4]) was developed by the Getty[5] foundation as a vehicle for indexing catalogues of art objects. Originally set up as a monolingual thesaurus in English, it is now also (partially) available in other languages, such as Dutch, Spanish and French. The AAT is widely used in musea and other cultural heritage institutions for cataloguing art collections. The AAT was developed according to the ISO standard for the definition of monolingual (ISO2788) and multilingual thesauri (ISO5964). These standards prescribe a data model which basically is a record structure with a number of attributes and three relations: hierarchical relation (broader/narrower term), equivalence of terms (synonyms and lexical variants)and an associative relation (related term).

```
LEN 513
STATUS n
IDNO 255420
DATCHG 19950712
DATENT 19950407
CN B.BM.CFS.AFU.ATG.RIQ.KKK.AHS
TERM allergies
ALT ALTERNATE: allergy
BT disease
SN SCOPE NOTE: Abnormal reactions of the body produced by a
        sensitizing dosage of or exposure to some foreign material.
HN April 1995 descriptor added.
SOURCE allergies (CCE; ROOT)
SOURCE allergy (CAND; MESH; OED2; RHUND2; W)
SOURCE allergic diseases (NASATH)
SOURCE allergy and immunology (MESH)
SOURCE hypersensitivity (MESH)
LINK allergy
```

Fig. 2. Example of the original AAT record representing the concept "allergies"

Fig. 2 shows an example of the AAT record representation of the concept "allergies". The record template of the AAT records is described in [6]. The field "IDNO" refers to a unique identifier of the entry. The "CN" field contains a code that determines the position of the term in the hierarchy. "TERM" and "ALT" contain the preferred term and alternative terms respectively. Besides "ALT", ATT uses also the fields "UF", "UK", "UKALT" and "UKUF" to indicate synonyms and alternative spellings. These field represent the equivalence realation

[4] ©2003, The J. Paul Getty Trust. All rights reserved.
[5] http://www.getty.edu/
[6] aat:usermanual

of the ISO standard, but are not alway applied consistently. "BT" refers to the broader term in the hierarchy. The "SN" field contains the scope note, a natural language description of the term. The example does not show the "RT" field which is used to represent related terms.

In step 1a of the conversion method the AAT record structure was converted by a simple Prolog program to the (partial) RDF(S) representation shown in Fig. 3. The mapping of the fields of the AAT record to an instance of the class AATTerm is generally straightforward. However, the coding of the `broaderTerm` field requires step 1b to convert the value of the record field BT, which is a term, to a unique reference (a `IDNO` to a concept. The mapping between the broader term and the identification number is simple in AAT since preferred terms are unique in AAT. An alternative way of determining the position of the entry in the hierarchy would be to use the value of the "CN" field (see also Sect. 5).

```
<aat:AATTerm rdf:about="&aat;255420">
  <aat:term>allergies</aat:term>
  <aat:alternate>allergy</aat:alternate>
  <aat:scopeNote>Abnormal reactions of the body produced by a
  sensitizing dosage of or exposure to some foreign material.
  </aat:scopeNote>
  <aat:broaderTerm rdf:resource="&aat;55130"/>
  <aat:source>allergy and immunology (MESH)</aat:source>
  <aat:source>hypersensitivity (MESH)</aat:source>
</aat:AATTerm>
```

Fig. 3. RDF(S) representation of (part of) the AAT record

Step 2a of the conversion procedure could involve the definition of certain relations between properties in a similar way as described in Sect. 3. In our current implementation this has not been done.

The representation as instances of the class AATTerm has only a limited meaning to RDF(S) knowledgable applications. In order to add subclass semantics to the instance representation (step 2b), we can make AATTerm a meta class and define the properties of the AAT record as subproperties of RDF(S) properties `rdfs:subClassOf`, `rdf:label` and `rdf:comment`, as is shown in Fig. 4.

These meta definitions allow the reinterpretation of the thesaurus entries as RDF(S) classes(i.e. instances of the meta-class AATTerm) and give the AAT properties a meaning which is interpretable within the semantics of RDF(S). For example the property "broaderTerm" is interpreted as a specialisation of the RDFS subClassOf relation resulting in a proper class hierarchy.

Since many thesauri are based on the same ISO2887 data model, the procedure described above can be applied in many cases. For example other resources of the Getty Foundation such as the ULAN [7] thesaurus of artist names and the

```
<rdfs:Class rdf:ID="&aat;AATTerm">
  <rdfs:subClassOf rdf:resource="&rdfs;Class"/>
</rdfs:Class>

<rdf:Property ID="&aat;broaderTerm">
  <rdfs:label>broader term</rdfs:label>
  <rdfs:domain rdf:resource="&aat;AATTerm"/>
  <rdfs:range rdf:resource="&aat;AATTerm"/>
  <owl:inverseOf rdf:resource="&aat;narrowerTerm"
  <rdfs:subPropertyOf rdf:resource="&rdfs;subClassOf"/>
</rdf:Property>

<rdf:Property ID="&aat;term">
  <rdfs:label>preferred term</rdfs:label>
  <rdfs:domain rdf:resource="&aat;AATTerm"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
  <rdfs:subPropertyOf rdf:resource="&rdfs;label"/>
</rdf:Property>

<rdf:Property ID="&aat;alternate">
  <rdfs:label>synonym</rdfs:label>
  <rdfs:domain rdf:resource="&aat;AATTerm"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
  <rdfs:subPropertyOf rdf:resource="&rdfs;label"/>
</rdf:Property>

<rdf:Property ID="&aat;scopeNote">
  <rdfs:label>scopenote</rdfs:label>
  <rdfs:domain rdf:resource="&aat;AATTerm"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
  <rdfs:subPropertyOf rdf:resource="&rdfs;comment"/>
</rdf:Property>
```

Fig. 4. Definitions of the AATTerm and its properties

thesaurus of geographical names TGN [6] which are available in record format can be easily converted to ontologies in a similar way as the AAT thesaurus.

5 Case 3: MeSH

The National Library of Medicine publishes the MeSH (Medical Subject Headings) thesaurus which provides a controlled vocabulary for indexing bio-medical literature. MeSH is available in a number of formats, including an XML format[7] [3]. Although we are aware of the fact that MeSH was not intended to be used as an ontology, we will demonstrate the conversion procedures using the XML representation of MeSH, since it reveals some important issues.

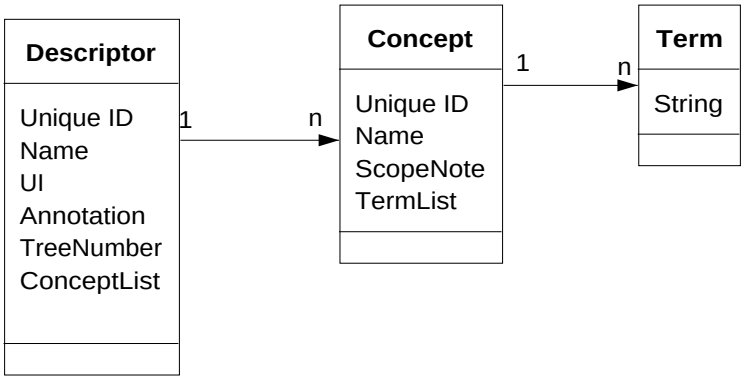

Fig. 5. The (simplified) data model of MeSH

A simplified version of the MeSH data model is shown in Fig. 5. An entry in MeSH is represented by a *descriptor* record that has a Unique Identifier, a Name, an optional Annotation and one or more TreeNumbers. The TreeNumber is a code that determines the position in the hierarchy of descriptors. Associated with a descriptor are one or more *concepts*. Concepts are used to represent sets of synonymous terms and scope notes. Concepts can have relations [2].

Fig. 6 shows an example of the XML representation of a descriptor record. The full XML representation of the MeSH descriptors is a large document (233 MB) so a streaming XML parser [8] is used to process the original data.

The first syntactic transformation from XML to RDF(S) (step 1a) involves the translation of the XML serialisation of the instances of the data model to instances of RDFS classes. Part of the RDF(S) schema used is shown in Fig. 7. Descriptors and concepts are modelled as instances of the classes `Descriptor` and `Concept` with attributes that correspond to the XML subelements. Since RDF(S) properties can have multiple values, the notions of ConceptList and TermList

[7] http://www.nlm.nih.gov/mesh/xmlmesh.html

```
<DescriptorRecord ...>                          <!-- Descriptor   -->
   <DescriptorUI>D000005</DescriptorUI>
   <DescriptorName><String>Abdomen</String></DescriptorName>
   <Annotation> region & abdominal organs...
   </Annotation>
   <ConceptList>

      <Concept PreferredConceptYN="Y">          <!-- Concept      -->
          <ConceptUI>M0000005</ConceptUI>
          <ConceptName><String>Abdomen</String></ConceptName>
          <ScopeNote> That portion of the body that lies
          between the thorax and the pelvis.</ScopeNote>
          <TermList>

             <Term ... PrintFlagYN="Y" ... >    <!-- Term         -->
                <TermUI>T000012</TermUI>
                <String>Abdomen</String>        <!-- String = the term itself -->
                <DateCreated>
                   <Year>1999</Year>
                   <Month>01</Month>
                   <Day>01</Day>
                </DateCreated>
             </Term>
             <Term  IsPermutedTermYN="Y"  LexicalTag="NON">
                 <TermUI>T000012</TermUI>
                 <String>Abdomens</String>
             </Term>
          <TermList>
      </Concept>
   </ConceptList>
</DescriptorRecord>
```

Fig. 6. Example MeSH descriptor record in XML

can be removed. The underlying assumption is that the order of the XML elements has no semantic significance (cf Sect. 2). In this stage the TreeNumber is simply stored as a string. The instances of the Term datatype are coerced to strings. This causes some loss of information (e.g. the date at which a term was created is lost), but this makes interpretation of the concepts in the ontology more transparent for the tools that we have currently available, such as Triple20 [**?**] (see also Fig. 9.

```
<rdfs:Class rdf:about="&mesh;Descriptor"
    rdfs:label="Descriptor">
  <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>

<rdfs:Class rdf:about="&mesh;Concept"
    rdfs:label="Concept">
  <rdfs:subClassOf rdf:resource="&mesh;mesh_root"/>
</rdfs:Class>

<rdf:Property rdf:about="&mesh;HasConcept"
    rdfs:label="HasConcept">
  <rdfs:domain rdf:resource="&mesh;Descriptor"/>
  <rdfs:range rdf:resource="&mesh;Concept"/>
</rdf:Property>

<rdf:Property rdf:about="&mesh;TreeNumber"
    rdfs:label="TreeNumber">
  <rdfs:domain rdf:resource="&mesh;Descriptor"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>

<rdf:Property rdf:about="&mesh;PharmacologicalAction"
    rdfs:label="PharmacologicalAction">
  <rdfs:domain rdf:resource="&mesh;Concept"/>
  <rdfs:range rdf:resource="&mesh;Descriptor"/>
</rdf:Property>
```

Fig. 7. Part of the RDF(S) schema for MeSH

In the second syntactic step (step 1a) the hierarchical relations that are implicit in the TreeNumbers are made explicit and modelled as a `subTreeOf` relation. In step 2b of the conversion of MeSH the same mechanism of meta-moddeling is used as for WordNet and AAT.

```
<rdf:Description rdf:about="&mesh;Descriptor">
  <rdfs:subClassOf rdf:resource="&rdfs;Class"/>
</rdf:Description>

<rdf:Property rdf:about="&mesh;subTreeOf"
    rdfs:label="subTreeOf">
  <rdfs:domain rdf:resource="&mesh;Concept"/>
  <rdfs:range rdf:resource="&mesh;Concept"/>
  <rdfs:subPropertyOf rdf:resource="&rdfs;subClassOf"/>
</rdf:Property>

<rdf:Property rdf:about="&mesh;ConceptTerm"
    rdfs:label="ConceptTerm">
  <rdfs:domain rdf:resource="&mesh;Concept"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
  <rdfs:subPropertyOf rdf:resource="&rdfs;label"/>
</rdf:Property>

<rdf:Description rdf:about="&mesh;DescriptorName">
  <rdfs:subPropertyOf rdf:resource="&rdfs;label"/>
</rdf:Description>

<rdf:Description rdf:about="&mesh;ConceptName">
  <rdfs:subPropertyOf rdf:resource="&rdfs;label"/>
</rdf:Description>

<rdf:Description rdf:about="&mesh;ScopeNote">
  <rdfs:subPropertyOf rdf:resource="&rdfs;comment"/>
</rdf:Description>
```

Fig. 8. The meta-schema definition of the MeSH ontology

6 Discussion and Conclusions

Ontologies are essential vehicles for the Semantic Web. Since RDF(S) and more recently OWL have become standard representation languages for ontologies the time has come to make the large variety of existing resources available for Semantic Web applications. The DAML repository of ontologies[8] is a first step towards this goal. However, the assumptions and methods that were used in creating the ontologies in this repository do not appear to be documented. The method presented in this paper supports the conversion of existing resources in such a way that the transformation steps can be made explicit and traceable. In addition, the method does not involve any changes in the original source material, the process consists just of mechanical transformation steps. This has the advantage that when new versions of the source material become available the conversion process can easily be repeated. The separation of the conversion process in syntactic and semantic steps allows for a gradual transition from a straightforward translation of the source to a semantic interpretation and augmentation of the source material. This has the advantage that a user can decide what transformations are acceptable for his or her purposes.

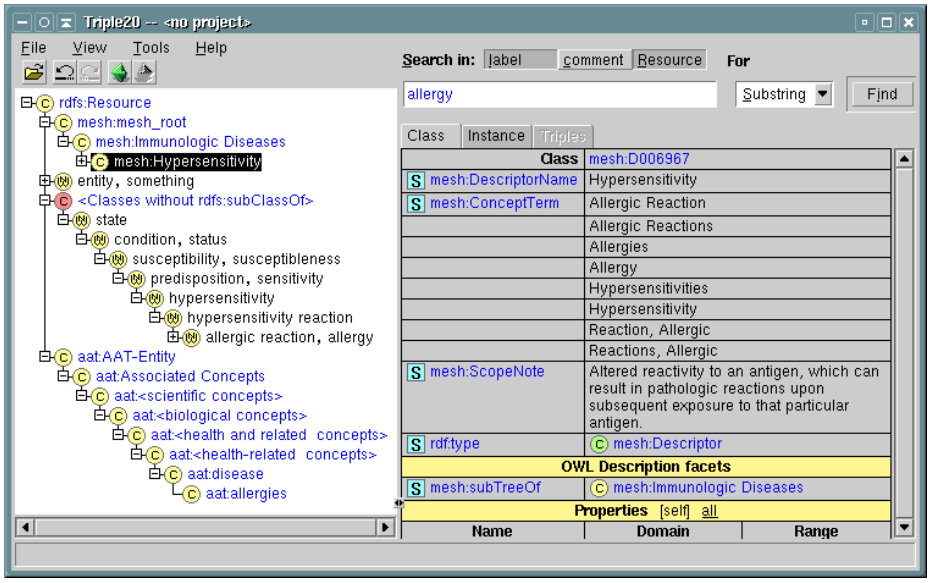

Fig. 9. Different representations of the concept "allergy" derived from three sources

Although the case studies described above are simplifications of the conversion process needed for a full mapping of the AAT and MeSH, they illustrate the principles of the method. An additional advantage of the methods is that the

[8] http://www.daml.org/ontologies

resulting ontologies can easily be compared using Semantic Web tools. Figure 9 shows a screenshot of the Triple20 ontology editer [9], [10] where an example concept ("allergy") is shown as it is represented in WordNet, AAT and MeSH. The uniform representation of the ontologies allows a comparative analysis of the different choices that were made in the different ontologies. A next step would be the mapping of equivalent or similar concepts from different ontologies. No doubt, such mappings will play an important role in future Semantic Web applications.

References

1. G. Miller. WordNet: A lexical database for english. *Comm. ACM*, 38(11), November 1995.
2. Stuart J. Nelson, Douglas Johnston, and Betsy L. Humphreys. *Relationships in Medical Subject Headings (MeSH)*, volume 2 of *Information Science and Knowledge Management*, chapter 11. Kluwer Academic Publishers, October 2001.
3. U.S. National Library of Medicine. Introduction to MeSH in XML format, November 2001.
4. T. Peterson. *Introduction to the Art and Architecture Thesaurus*. Oxford University Press, 1994. See also: http://www.getty.edu/research/tools/vocabulary/aat/.
5. A. Th. Schreiber, B. Dubbeldam, J. Wielemaker, and B. J. Wielinga. Ontology-based photo annotation. *IEEE Intelligent Systems*, 16(3):66–74, May/June 2001.
6. TGN: Thesaurus of Geographical Names. The Getty Foundation. URL: http://www.getty.edu/research/tools/vocabulary/tgn/, 2000.
7. ULAN: Union List of Artist Names. The Getty Foundation. URL: http://www.getty.edu/research/tools/vocabulary/ulan/, 2000.
8. J. Wielemaker. *SWI-Prolog SGML/XML parser*. SWI, University of Amsterdam, Roetersstraat 15, 1018 WB Amsterdam, The Netherlands, 2002. URL: http://www.swi-prolog.org/packages/sgml2pl.html.
9. J. Wielemaker. *Triple20 – An RDF/RDFS/OWL visualisation and editing tool*. SWI, University of Amsterdam, Roetersstraat 15, 1018 WB Amsterdam, The Netherlands, 2003. URL: http://www.swi-prolog.org/packages/Triple20.html.
10. Jan Wielemaker, Guus Schreiber, and Bob Wielinga. Prolog-based infrastructure for RDF: Scalability and performance. In J. Mylopoulos D. Fensel, K. Sycara, editor, *The Semantic Web-ISWC2003*, pages 644–658, Berlin Heidelberg, 2003. Springer. LNCS 2870.