Representing the Hierarchy of Industrial Taxonomies in OWL: The gen/tax Approach

Martin Hepp

Digital Enterprise Research Institute (DERI), University of Innsbruck
Florida Gulf Coast University, Fort Myers, FL, USA
martin.hepp@deri.org

Abstract. Existing taxonomies are valuable input for creating ontologies, because they reflect some degree of community consensus and contain, readily available, a wealth of concept definitions plus a hierarchy. However, the transformation of such taxonomies into useful ontologies is not as straightforward as it appears, because simply taking the hierarchy of concepts, which was originally developed for some external purpose other than ontology engineering, as the subsumption hierarchy using rdfs:subClassOf can yield useless ontologies. In this paper, we (1) illustrate the problem by analyzing OWL and RDF-S ontologies derived from UNSPSC (a products and services taxonomy), (2) detail how the interpretation and representation of the original taxonomic relationship is an important modeling decision when deriving ontologies from existing taxonomies, (3) propose a novel "gen/tax" approach to capture the original semantics of taxonomies in OWL, based on the split of each category in the taxonomy into two concepts, a *generic concept* and a *taxonomy concept*, and (4) show the usefulness of this approach by transforming eCl@ss into a fully-fledged products and services ontology.

Keywords. Ontology engineering, OWL, reuse, taxonomies, UNSPSC, eCl@ss, e-business

1. Introduction

Standard taxonomies exist in different problem domains and contain many concept definitions plus a hierarchy. UNSPSC, a standard taxonomy for products and services and often referred to as a business ontology, contains 20,789 categories (in version 7,0901), and eCl@ss, a similar but more expressive standard, contains 25,658 categories plus 5,525 precisely defined object and datatype properties (in version 5.1de). For a quantitative analysis of the content and domain coverage of products and services taxonomies, see [1] and [2]. However, the transformation of such taxonomies into useful ontologies is not as straightforward as it appears, because simply taking the hierarchy of concepts, which was originally developed for some external purpose other than ontology engineering, as the subsumption hierarchy using rdfs:subClassOf can yield useless ontologies. Often, the original meaning of the taxonomic relationship is *"A is in some context a more specific category of B"* rather than a strict subClassOf relationship with the typical semantics *"For all A being a subordinate node of B, every instance of A shall*

also be an instance of B". UNSPSC, for example, treats "ice" as a subordinate node of "beverages", and eCl@ss has "docking stations" as a subordinate node of "computers".

Our paper is related to the following previous works: First, the analysis of the meaning of taxonomic relationships, especially the fundamental work of [3]. Second, methodologies for and experiences with the reuse of consensus in existing standards for the creation of ontologies. This is the most relevant field of work for this paper. [4] discusses the transformation of tangled hierarchies, as e.g. such derived from ambiguous "broader than / narrower than" taxonomies in library science, into formal ontologies. [5] presents the experiences gained while transforming the constructs of an existing semantic net in the medical domain into an OWL ontology. [6] is a detailed description of creating products and services ontologies based on UNSPSC and eCl@ss. [7] shows the reuse and semantic enrichment of an existing taxonomy, and demonstrates this for the Art and Architecture Thesaurus (AAT). [8] and [9] are consequent works of this stream of research.

The structure of the paper is as follows: In section 2, we describe the problem of representing the hierarchy of existing taxonomies in derived ontologies. In section 3, we propose a novel "gen/tax" approach, based on the separation of generic and taxonomy concepts. In section 4, we discuss this approach and show how it can be successfully applied to the representation of eCl@ss in OWL. Section 5 concludes the paper.

2. Representing the Hierarchy of Taxonomies in OWL and RDF-S

In this section, we show that in informal taxonomies, the meaning of the taxonomic relationship and the intension of the concepts are tangled, and that the interpretation and representation of the hierarchy of a given taxonomy in an ontology language is an important modeling decision that affects the usefulness of the resulting ontology.

When taking the categories found in a taxonomy as the basis for the creation of an ontology, we face a fundamental problem: Unless there is a formal definition of the semantics of the taxonomic relationship, the intensions of the category concepts (e.g. the product classes) *are not determined independently from the interpretation of the taxonomic relationship.* In other words: If we lack a formal definition of either the hierarchical relationships or the category concepts, then *how we understand the taxonomic relationship determines the shape of the category concepts* and vice versa. Our choice of the interpretation of the taxonomic relationship affects the intension of the category concepts, and a chosen definition of the intension of the category concepts is compatible with only a specific interpretation of the taxonomic relationship. As a consequence, we have

some degree of choice over the intension of the ontology classes derived from the categories in the source taxonomy by selecting the interpretation of the taxonomic relationship.

Two examples might illustrate this fundamental problem: The hierarchies of both UNSPSC and eCl@ss were created on the basis of practical aspects of procurement, treating those commodities that "somehow" belong to a specific category, as descendents of this closest category. This makes "ice" a subclass of "non-alcoholic beverages" in UNSPSC and "docking stations" a subcategory of "computers" in eCl@ss. Now, we still can read the taxonomic relationship as a strict "rdfs:subClassOf" relationship (i.e. each instance of "ice" is also an instance of "non-alcoholic beverages" and each instance of "docking station" is also an instance of "computers"). Then, however, the intension of the class "computers" is no longer any computer, but the concept "computer" solely from the perspective of cost accounting or spend analysis, where an incoming invoice for a docking station can be treated as an incoming invoice for a computer. Similarly will "non-alcoholic beverages" no longer represent all non-alcoholic beverages, but the union of non-alcoholic beverages and related commodities. The problem arises only because we have to narrow down the semantics of the original informal standard when turning it into an ontology. Basically, each source taxonomy contains two concepts for each category node: First the generic concept of the respective category (e.g. "computer") and second the intersection of this concept with a concept "element in this taxonomy", the latter reflecting all the implicit assumptions of the creators of the taxonomy and the constraints resulting from the interpretation of the taxonomic relationship.

The most straightforward approach to deal with this is to define a transitive relationship "taxonomySubClassOf" that can be used to represent the original hierarchy in the taxonomy. Then, the original hierarchy would not be falsely used as the subsumption hierarchy, turning instances of "ice" into instances of "beverages", but would still be available for queries. However, this is not possible in neither OWL Lite, OWL DL, nor RDF-S. In OWL Lite and OWL DL, it is not possible to define a transitive relation that links classes; such can only be annotation properties that cannot be assigned a formal semantics. In RDF-S, it is not possible to define transitivity of properties.

There are at least the following three approaches of transforming a given taxonomy into an OWL Lite or DL ontology:

1. Create one class for each taxonomy category and assume that the meaning of the taxonomic relationship is equivalent to rdfs:subClassOf.
2. Create one class for each taxonomy category and represent the taxonomic relationship using an *annotation* property taxonomySubClassOf in OWL.
3. Treat the category concepts as instances instead of classes and connect them using a transitive object property taxonomySubClassOf.

In RDF-S, only the first two alternatives are possible, with taxonomySubClassOf being a regular RDF property for solution 2. Approach 1 is chosen by both available transformations of UNSPSC into products and services ontologies [10] [11]. Solution 2 seems to be the most straightforward alternative, since the specific meaning of the taxonomic relationship is captured using a specific property, and the classes can represent generic product concepts. The problem with this approach is that, in OWL Lite and OWL DL, a property that connects classes with classes can only be an annotation property. Thus, it cannot be made a transitive property, and an OWL Lite or OWL DL reasoner will only see explicit statements. In other words, if class A is a taxonomy-SubClassOf of class B and class B is a taxonomySubClassOf of C, then the reasoner will not infer that class A is also a taxonomy-SubClassOf of class C. This limitation can be avoided by making the products and services concepts instances instead of classes, as described in solution 3. Then, the property "taxonomySubClassOf" can be an owl:ObjectProperty and can be made transitive. The downside of this approach is that one absolutely needs OWL Lite or OWL DL reasoning support in order to process the transitive nature of the property. We also think that it is not very intuitive to model categories, which are meant as abstract classes, in the form of instances.

3. The gen/tax Approach

In this section, we describe a fourth approach of transforming existing taxonomies into domain ontologies. Our approach is based on the idea of deriving *two* concepts for each taxonomy category, one reflecting the *generic* concept and another reflecting the *taxonomy* concept. The advantage of this solution is that it works with the "intersection" of RDF-S and OWL Lite, i.e. it does not require reasoning capabilities beyond rdfs:subClassOf. This limited requirement on the ontology language has also advantageous effects from an implementation perspective.

The basic idea of this "gen/tax" approach is as following:

1. We create two separate concepts for (1) the *generic* category and (2) the respective *taxonomy* category.
2. We arrange the *taxonomy* concepts in a rdfs:subClassOf hierarchy that is identical to the original order in the taxonomy, but don't do this for the generic concepts. The generic concepts are just named classes without any support for subsumption, but can be manually augmented at a later point in time. This split allows for capturing the hierarchy of taxonomy concepts without linking the generic concepts to incorrect superordinate classes.

3. In order to ease annotation, we create one "annotation" class for each taxonomy node, which becomes an rdfs:subClassOf of *both* the respective generic and the respective taxonomy concept. With this construct, a single rdf:type statement is sufficient to make a resource an instance of both the generic and the taxonomy concept.

In the following, we illustrate the gen/tax approach using an example. The example is based on the sample products and services taxonomy shown in Figure 1.

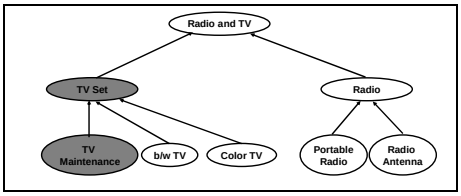

Fig. 1. Example of a products and services taxonomy

Fig. 2 illustrates how the two grey-shaded categories in Fig. 1 can be represented using the gen/tax approach. The generic concept "TV Set" represents all TV Sets. The taxonomy concept "TV Set" represents "TV Set"-related instances in the context of the original taxonomy (*"everything that shall be treated as a TV Set in the context of the ordering purpose of the original taxonomy"*). The annotation concept, being a subclass of both, is just for convenience reasons and allows making an instance, e.g. the concrete TV set model "sony:TV-123" an instance of both the generic and the taxonomy concept with one single rdf:type statement.

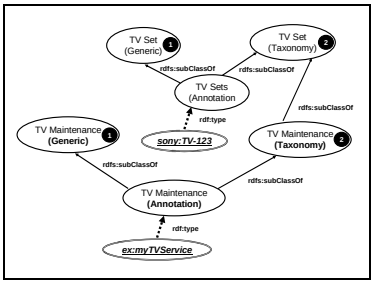

Fig. 2. The gen/tax approach: Separating the generic concept from the taxonomic concept

This approach allows preserving the original order of the taxonomy without narrowing down the intensions of the concepts to one application domain, and puts only minimal requirements on the expressiveness of the ontology language. One

might argue that the generic classes are of little value, because they are just named classes. However, there are two counterarguments: First, more semantics cannot be automatically deduced from the input taxonomy. It might be desirable to infer more about these classes, but this means manual ontology engineering work. Keep in mind that the taxonomies under discussion can be rather big with more than 20,000 concepts and are also volatile with multiple releases per year. Second, in combination with a library of properties, already a flat set of consensual, named classes can bring a lot for the application domain, e.g. in e-business.

The application of this approach for describing products is shown in Fig. 3: The TV maintenance service ex:tv-set-repair is an instance of the annotation class "TV Set Maintenance". This makes it also an instance of the generic product class "TV Set Maintenance (Generic)" and the taxonomy concept "TV Set Maintenance (Taxonomy)". The second is a subclass of "TV Set (Taxonomy)", but the first is not a subclass of "TV Set (Generic)".

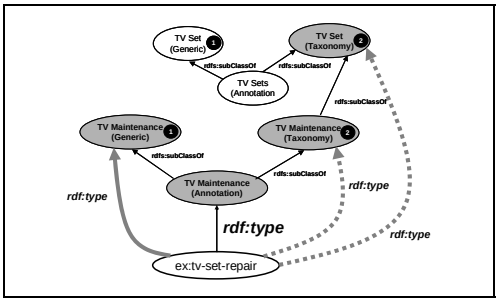

Fig. 3. Usage of the gen/tax ontology for product description

This yields exactly the distinction we want: When searching for a TV maintenance service, we look for instances of the *generic* class, and when looking for all items that belong to the taxonomy category, we use the taxonomy concept. For example, a store manager might want to find all products in the TV set segment. In this case, he or she also wants to find TV set cabling and maintenance, so the query will be based on the taxonomy concept. The taxonomic concepts on the right side are the narrow interpretations of the taxonomic categories and are arranged in the original hierarchical order. The generic concepts on the left side are the original, broad variants of the taxonomy concepts, not constrained by the implications of reading the taxonomic relationships as rdfs:subClassOf. The convenience classes in the middle ease the annotation task, since an instance can be made an instance of both the generic and the taxonomy concept by just one rdf:type statement.

4. Validation and Discussion

As a validation of the gen/tax approach, we transformed the latest version of eCl@ss into a fully-fledged products and services ontology, called eClassOWL. The project will be available at http://www.heppnetz.de/eclassowl. In the following, we give a short example of the usage of the resulting ontology for product description in the Semantic Web. We assume that "Fendt Supermower" is an agricultural machine (eCl@ss category AKK255001), its weight is 125.5 kg, and the manufacturer name is "Fendt". As concept identifiers, we use the primary key (e.g. "AAA001001") plus a prefix for classes and properties ("C_", and "P_"). The annotation class has the resulting string as its concept identifier. The generic class has an additional "-gen" and the taxonomic class has an additional "-tax". In other words, the eCl@ss category "agricultural machine" (primary key "AKK255001") is represented using the following three concepts: (1) **C_AKK255001** for the annotation concept, (2) **C_AKK255001-gen** for the generic concept, and (3) **C_AKK255001-tax** for the taxonomic concept. Assumed that the ID for this product instance is "machine1", the respective product description using the eCl@ss ontology would be as follows:

```
<pcs:C_AKK255001 rdf:ID="machine1">
<pcs:P_AAA042001>125.50</pcs:P_AAA042001><!-- Weight -->
<pcs:P_AAA001001>Fendt</pcs:P_AAA001001><!-- Manufacturer -->
<pcs:P_AAA003001>Fendt Supermower1234</pcs:P_AAA003001><!-- Name -->
</pcs:C_AKK255001>
```

Now, we want to search for all agricultural machines in the ontology that weigh less than 160 kg. The respective RDQL query would be:

```
SELECT ?x, ?weight, ?productName, ?vendor WHERE
   (?x, <rdf:type>, <pcs:C_AKK255001-gen>)
   (?x, <pcs:P_AAA001001>, ?vendor)
   (?x, <pcs:P_AAA003001>, ?productName)
(?x, <pcs:P_AAA042001>, ?weight)
AND ?weight <160
```

Because we want to get only instances of the generic product category, the class to be used in the query is C_AKK255001-**gen**, not C_AKK255001-**tax**. The later could be used to determine all products that fall in the respective taxonomy category. For example, a store manager might want to see all products in this product segment, including maintenance and spare parts for agricultural machines. Just changing the class ID in the query to C_AKK255001-**tax** would return exactly that.

5. Conclusion

In this paper, we have proposed a novel "gen/tax" approach that allows the mechanized transformation of taxonomies into OWL or RDF-S ontologies based

on the representing the original taxonomy node in two concepts, one for the generic concept and one for the taxonomic concept, plus a third convenience class for easing annotation. We have applied our approach to the task of creating an industry-strength products and services ontology based on eCl@ss. The approach has the disadvantage that it increases the number of classes in the resulting ontology. We do not think that this is a significant problem, especially since performance-wise, this will likely be outweighed by possibility to use a very simple and more scalable reasoner. It is noteworthy that ontology languages without the limitations of OWL or RDF-S allow a more straightforward modeling of the original semantics by just creating a transitive relationship "taxonomySubClassOf". This is especially true for all variants of WSML and WRL. Still it is important to observe the fact that the interpretation of the taxonomic relation in informal taxonomies is a crucial modeling decision when deriving ontologies from such existing taxonomies. In WSML and WRL, our findings can be represented without the OWL/RDF-S-specific workaround of separate classes per each category in the taxonomy.

Acknowledgements: The work presented in this paper is partly funded by the European Commission under the project DIP (FP6-507483), the TransIT Entwicklungs- und Transfercenter at the University of Innsbruck, and Florida Gulf Coast University.

References

[1] M. Hepp, J. Leukel, and V. Schmitz, "Content Metrics for Products and Services Categorization Standards", presented at IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE-05), Hong Kong, 2005.

[2] M. Hepp, J. Leukel, and V. Schmitz, "A Quantitative Analysis of eCl@ss, UNSPSC, eOTD, and RNTD Content, Coverage, and Maintenance", accepted for IEEE ICEBE 2005, Beijing, China, 2005.

[3] R. J. Brachman, "What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks", *IEEE Computer*, vol. 16, pp. 30-36, 1983.

[4] A. L. Rector, C. Wroe, J. Rogers, and A. Roberts, "Untangling Taxonomies and Relationships: Personal and Practical Problems in Loosely Coupled Development of Large Ontologies", presented at K-CAP'01, Victoria, British Columbia, Canada, 2001.

[5] V. Kashyap and A. Borgida, "Representing the UMLS Semantic Network using OWL", presented at 2nd International Semantic Web Conference 2003 (ISWC 2003), Sanibel Island, Florida, USA, 2003.

[6] M. Hepp, "A Methodology for Deriving OWL Ontologies from Products and Services Categorization Standards", presented at the 13th European Conference on Information Systems (ECIS2005), Regensburg, Germany, 2005.

[7] B. J. Wielinga, A. T. Schreiber, and J. A. C. Sandberg, "From Thesaurus to Ontology", presented at First International Conference on Knowledge Capture (K-CAP 2001), Victoria, British Columbia, Canada, 2001.

[8] B. J. Wielinga, J. Wielemaker, G. Schreiber, and M. van Assem, "Methods for Porting Resources to the Semantic Web", *Proceedings of the First European Semantic Web Symposium (ESWS'04), Heraklion, Greece*, pp. 299-311, 2004.

[9] M. van Assem, M. R. Menken, G. Schreiber, J. Wielemaker, and B. J. Wielinga, "A Method for Converting Thesauri to RDF/OWL", presented at ISWC'04, Hiroshima, Japan, 2004.

[10] M. Klein, "DAML+OIL and RDF Schema representation of UNSPSC", available at http://www.cs.vu.nl/~mcaklein/unspsc/.

[11] D. L. McGuinness, "UNSPSC Ontology in DAML+OIL", available at http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml.