Poster Proposal for ISWC 2005

eClassOWL: A Fully Fledged Products and Services Ontology in OWL

Martin Hepp
Digital Enterprise Research Institute (DERI)
University of Innsbruck
Technikerstraße 21 a, 6020 Innsbruck, Austria
martin.hepp@deri.org

Abstract

A major obstacle towards e-Commerce applications based on Semantic Web technology is the lack of industry-strength ontologies for products and services. Besides simple, script-based transformations of the UNSPSC taxonomy, there are currently no serious ontologies for products and services. On this poster, we show that there exist several pitfalls when reusing industrial products and services taxonomies, namely UNSPSC and eCl@ss, as the input for a products and services ontology for eCommerce. We especially show that the representation of the semantics of the original taxomic relationship is an important modeling decision, which eventually affects the usefulness of the resulting ontology. As a demonstration of our findings, we describe eClassOWL, which is an industry-strength ontology for products and services based on eCl@ss.

1. Introduction

A major obstacle towards e-Commerce applications based on Semantic Web technology is the lack of industry-strength ontologies for products and services. Besides very simple, script-based transformations of the UNSPSC taxonomy, there are currently no serious ontologies for products and services.

Numerous researchers have pointed out the enormous potential of ontologies for e-business scenarios, for example [1-8], and many sub-problems of making this a reality have been addressed, e.g. the challenges associated with mapping among e-business ontologies [5].

In our analysis we show how useful products and services ontologies can be derived from UNSPSC and eCl@ss in a script-based fashion.

1.1. Related Work

Related work can be classified into the following main groups:

- Analysis of the meaning of taxonomic relationships, especially the fundamental work of [3]. This yielded the insight that there are multiple types of taxonomic relationships, which should be represented separately.
- Ontology engineering methodologies, implicitly or explicitly focusing on the manual creation of ontologies based on knowledge engineering principles. A comprehensive discussion of all approaches in this field is beyond the scope of this paper, for an overview see e.g. [4] and [5].
- 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 our work. [6] discusses the transformation of tangled hierarchies, as e.g. such derived from ambiguous “broader than / narrower than” taxonomies in library science, into formal ontologies. [7] presents the experiences gained while transforming the constructs of an existing semantic network in the medical domain into an OWL ontology. [8] is our detailed description of creating products and services ontologies based on UNSPSC and eCl@ss. [9] shows the reuse and semantic enrichment of an existing taxonomy, and demonstrates this for the Art and Architecture Thesaurus (AAT). [10] and [11] are consequent works of this stream of research. An important characteristic of [10] and [11] is that the authors leave the limits of OWL DL in order to capture semantics contained in the original taxonomy, namely to be able to treat classes as instances and vice versa.

- Prototypes of products and services ontologies in standard ontology languages derived from UNSPSC. To our knowledge, there are currently two examples of UNSPSC transformations into ontology representation languages: The DAML+OIL and RDF-S variants created by [12] and the DAML+OIL variant from the Knowledge Systems Laboratory at Stanford University [13].

1.2. Our Contribution

As a result of our work we show that (1) the interpretation and consequent representation of the semantics of the original taxomic relationship of UNSPSC and eCl@ss is an important modeling decision, which eventually affects the usefulness of the resulting ontology, (2) the original semantics cannot be directly represented in OWL and RDF-S. As a consequence, we (3) propose a generic workaround for this type of problem. As a demonstration of our findings, we (4) describe eClassOWL, which is an industry-strength ontology for products and services based on eCl@ss.

2. Ontology Engineering Challenges

The basic challenge when deriving ontologies from product standards is to represent as much as possible of the original semantics in the taxonomy while using the allowed constructs of the respective ontology language. In this section we explain how previous attempts fail to properly reflect the semantics of the reused products standards, especially with regard to the taxonomic relationship, and propose a novel methodology.

2.1. Product Classes and Hierarchy

When taking the categories found in a taxonomy as the basis for the creation of an ontology, we face a fundamental -- but so far ignored -- 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 fatal consequence of this interpretation of the taxonomic relationship as being equivalent to rdfs:subClassOf is obvious: We can no longer use the resulting classes for buying processes, because a search for all instances of "computers" will also return docking stations, and ordering the cheapest available instance of non-alcoholic beverages will very likely return just ice cubes.

Given now the fact that our interpretation of the taxonomic relationship in the source taxonomy determines the intension of the classes, we have to *deliberately select* the most useful shape of the product classes and then derive the required interpretation for the taxonomic relationship. Basically, each source taxonomy contains two concepts for each category node: First the generic concept of the respective product or service category (e.g. "computer", marked as (1) in Figure 1) and second the intersection (marked as (2) in Figure 1) of this concept with a concept "element in this taxonomy" (marked as (3) in Figure 1), the later reflecting all the implicit assumptions of the creators of the taxonomy and the constraints resulting from the interpretation of the taxonomic relationship.

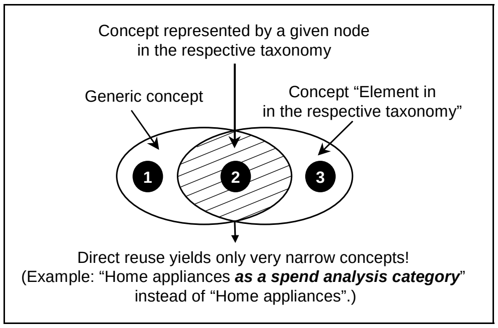

Figure 1: Relationship between generic concepts and taxonomy nodes

Now, even though the taxonomic relationship should not be interpreted as rdfs:subClassOf, we still want to capture the relationship as such, because there are applications like cost accounting where it is very useful. So we have to find a way to represent both generic categories for general use, while still preserving the hierarchical order for analytical purposes and similar applications.

Basically there are at least the following four approaches of transforming a given products taxonomy into an OWL Lite 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.
3. Treat the category concepts as instances instead of classes and connect them using a transitive object property taxonomySubClassOf.
4. Create two concepts for each taxonomy category, one reflecting the generic product or service type and another reflecting the taxonomy concept.

Approach 1 is chosen by both available transformations of UNSPSC into products and services ontologies. For example, the RDF Schema representation of UNSPSC created by [13] contains statements of the following kind:

```
<rdfs:Class rdf:ID="Ice">
  <rdf:type
rdf:resource="http://ontoview.org/schema/unspsc/1#
Commodity"/>
  <rdfs:subClassOf
rdf:resource="#Non_alcoholic_beverages"/>
  <unspsc:egci>014067</unspsc:egci>
  <unspsc:code>50.20.23.02</unspsc:code>
</rdfs:Class>
```

The DAML representation of UNSPSC created by the Knowledge Systems Laboratory at Stanford University [14] uses the same structure.

While the underlying approach is not necessarily incorrect, it does not yield a products and services ontology, but a set of cost accounting and purchasing management categories. Quite clearly, we want to make the resulting products and services ontology be useful for many different application areas, including the search for products and services, and not limit the usage to spend analysis.

Solution 2 seems to be the most straightforward alternative, since the specific meaning of the taxonomic relationships 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 a OWL Lite or OWL DL reasoner will only see explicit statements. In other words, if class A is a taxonomySubClassOf class B and class B is a taxonomySubClassOf C, the reasoner will not infer that class A is also a taxonomySubClassOf 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. For scalability reasons, we wanted to find a solution that works with the "intersection" of RDF-S and OWL Lite, i.e. that does not require reasoning capabilities beyond rdfs:subClassOf..

This leaves solution 4, which we currently favor. The basic idea is as following:

1. We create two separate concepts for (1) the *generic* product or service category and (2) the respective *taxonomy* category.
2. We arrange the *taxonomy* concepts in a strict rdfs:subClassOf hierarchy, but not the generic concepts. This 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 *both* the respective ge-

neric and the respective taxonomy concept. With this construct, a single rdf:type statement is sufficient to make a product an instance of both the generic and the taxonomy concept.

Figure 2 illustrates this approach. The concept numbering with (1) and (2) refers to the numbering in Figure 1. 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.

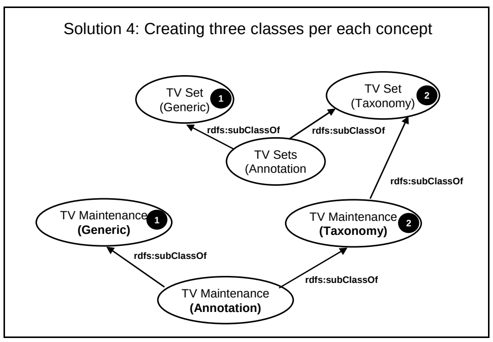

Figure 2: Alternate solution: Separating the generic concept from the taxonomic concept

The resulting ontology is not limited to describing product instances, but can also be used for annotating any kind of other entities. For example, incoming invoices can be linked to the respective classes with an annotation property "ex:belongsToCategory". This allows capturing the fact that a specific invoice is referring to the respective category, but is not an instance. As every class in OWL is also an rdfs:subClassOf itself and rdfs:subClassOf is transitive, one can easily search for all resources belonging to a taxonomy category on any level of the hierarchy.

2.2. Product Properties and Property Data Typing

From an ontology engineering perspective, the set of properties in product standards might seem rather trivial, because most of them are simple data type properties. However, their contribution to machine readable semantics in the Semantic Web is huge, for they provide standardized representation for concepts as generic as "weight" or as specific as "pump capacity". Some of them can also be applied usefully when the class of a product is not known or no proper class exists. For example, it might be helpful to represent the weight of a novel product even though we do not yet have a proper class for this product.

The import of properties from PSCS into an OWL Lite ontology requires the following steps:

1. Determine whether it is a data type property or an object property, i.e. whether there are enumerated data values for this property or not.
2. Map the data type of the property to one XSD data type supported by current reasoners.
3. Add the unit of measurement (e.g. "inches"), usually stored separately in the PSCS, to the description of the property.
4. Create respective OWL properties for each property contained in the source PSCS.
5. Create instances of a new class "PropertyValue" for each enumerated data value.

It is important to know that the data types used in the many standards cannot be directly mapped to standard XSD data types. eCl@ss, for example, uses over hundred different data type definitions based on ISO 9735 and ISO 6093, which are far more specific than available standard XSD data types.

3. eClassOWL: A Real Products and Services Ontology in OWL

In the following section, we give an example of how to use the resulting eClassOWL ontology for products and services representation. The following implementation details are important in order to understand the consequent demonstration:

We use the primary key of all taxonomy elements (e.g. "AAA001001") as concept identifiers, for they guarantee uniqueness. Since there are overlaps in the primary keys of properties and classes, we added 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.

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". 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.

4. Status of the Project

We have finished the transformation of eClass 5.1 into a products and services ontology. Currently, we are preparing the legal framework and documentation. The final version will be available at

http://www.heppnetz.de/eclassowl.

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

References

[1] D. Fensel, *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Berlin etc.: Springer, 2001.

[2] D. Fensel, Y. Ding, B. Omelayenko, E. Schulten, G. Botquin, M. Brown, and A. Flett, "Product Data Integration in B2B E-Commerce," *IEEE Intelligent Systems*, vol. 16, pp. 54-59, 2001.

[3] D. Fensel, D. L. McGuinness, E. Schulten, W. K. Ng, E.-P. Lim, and G. Yan, "Ontologies and Electronic Commerce," *IEEE Intelligent Systems*, vol. 16, pp. 8-14, 2001.

[4] T. Gupta and A. Qasem, "Reduction of price dispersion through Semantic E-commerce: A Position Paper." Hawai, USA, 2002.

[5] L. Obrst, R. E. Wray, and H. Liu, "Ontological Engineering for B2B E-Commerce," presented at International Conference on Formal Ontology in Information Systems (FOIS'01), Ogunquit, Maine, USA, 2001.

[6] B. Omelayenko, "Integration of Product Ontologies for B2B Marketplaces: A Preview," *ACM SIGecom Exchanges archive*, vol. 2, pp. 19-25, 2000.

[7] B. Omelayenko, "Ontology Integration Tasks in Business-to-Business E-commerce," presented at Fourteenth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, June 4-7, 2001, Budapest, Hungary, 2001.

[8] Y. Zhao, "Develop the Ontology for Internet Commerce by Reusing Existing Standards," *Proceedings of International Workshop on Semantic Web Foundations and Application Technologies (SWFAT), Nara, Japan, March 12, 2003*, pp. 51-57, 2003.

[9] E. Schulten, H. Akkermans, G. Botquin, M. Dörr, N. Guarino, N. Lopes, and N. Sadeh, "The E-Commerce Product Classification Challenge," *IEEE Intelligent Systems*, vol. 16, pp. 86-89, 2001.

[10] A. M. Fairchild and B. de Vuyst, "Coding Standards Benefiting Product and Service Information in E-Commerce," *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS-35)*, pp. 258b, 2002.

[11] D. Kim, J. Kim, and S.-g. Lee, "Catalog Integration for Electronic Commerce through Category-Hierarchy Merging Technique," *Proceedings of the 12th International Workshop on Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems (RIDE'02)*, pp. 28-33, 2002.

[12] D. Kim, S.-g. Lee, J. Chun, and J. Lee, "A Semantic Classification Model for e-Catalogs," *Proceedings of the IEEE International Conference on E-Commerce Technology*, 2004.

[13] M. Klein, "DAML+OIL and RDF Schema representation of UNSPSC," vol. 2004. Amsterdam, 2002.
[14] D. L. McGuinness, "UNSPSC Ontology in DAML+OIL," vol. 2004, 2001.
[15] W3C, "XML Schema Part 2: Datatypes Second Edition. W3C Recommendation 28 October 2004," vol. 2004, 2001.