A Case Study in Integrating Multiple E-commerce Standards via Semantic Web Technology

Yang Yu, Donald Hillman, Basuki Setio, and Jeff Heflin

Department of Computer Science and Engineering. Lehigh University
19 Memorial Drive West, Bethlehem, PA 18015
{yay208,djh3,bas207,heflin}@cse.lehigh.edu

Abstract. Internet business-to-business transactions present great challenges in merging information from different sources. In this paper we describe a project to integrate four representative commercial classification systems with the Federal Cataloging System (FCS). The FCS is used by the US Defense Logistics Agency to name, describe and classify all items under inventory control by the DoD. Our approach uses the ECCMA Open Technical Dictionary (eOTD) as a common vocabulary to accommodate all different classifications. We create a semantic bridging ontology between each classification and the eOTD to describe their logical relationships in OWL DL. The essential idea is that since each classification has formal definitions in a common vocabulary, we can use subsumption to automatically integrate them, thus mitigating the need for pairwise mappings. Furthermore our system provides an interactive interface to let users choose and browse the results and more importantly it can translate catalogs that commit to these classifications using compiled mapping results.

Keywords: e-commerce, ontology integration.

1 Introduction

Internet business-to-business transactions afford both dramatically increased flexibility and present great challenges in merging information coming from so many sources. To provide B2B services, suppliers must deal with the problem of heterogeneity underlying their customers' product, catalog, and document descriptions. Due to the lack of global common standards to classify products, a key task for these marketplaces is to effectively and efficiently manage different description styles. In real-world marketplaces, developing a scalable approach for information integration has become essential to expanding business.

The Federal Cataloging System (FCS) is an extensive taxonomy for naming, classifying, and describing items of supply. The Defense Logistics Information Service (DLIS) created the FCS and uses it to catalog and differentiate items of supply. DLIS routinely catalogs a varied population of items. The conventional coding of products for internal operations has resulted in product identifiers that

A. Bernstein et al. (Eds.): ISWC 2009, LNCS 5823, pp. 909–924, 2009.

are not easily exchangeable across taxonomies, causing significant duplication and inefficiencies in business processes. For DLIS to have full access to suppliers and their products everywhere and at all times, the FCS taxonomy must be aligned with supplier taxonomies.

There are many taxonomies and classification schemes and more than 40 of them have been publicly identified. Most of them organize data into domains viewed from the perspective of a specific use case. ECl@ss is a Standardized Material and Service Classification, while Product Service Classification (PSC), United Nations Standardized Products and Services Code (UNSPSC), Common Procurement Vocabulary (CPV), Common Procurement Code (CPC), RosettaNet Technical Dictionary (RTD), and the Customs Harmonized Tariff Code (HS) are commodity classifications. It is not only products that appear in taxonomies. For example, the Standard Industrial Classification (SIC) code and the North American Industrial Classification Code (NAIC) organize companies by principal occupation, such as Manufacturer, Wholesaler, Retailer, etc.

In this project we integrated four representative commercial classification systems with the FCS by focusing on their underlying knowledge structures rather than on their surface characteristics. Since these four systems adopt significantly differing classification strategies and cover most leading classification styles, we believe that the experience of this development is sufficiently general and capable of being used in most other popular classifications. The four classifications we used were the UNSPSC, CPV, eCl@ss, and the ISO 13584-511 Fasteners Dictionary. (The official acronym for ISO 13584 is PLIB, so we use PLIB-511 as its short name in this paper.)

The paper is organized as following. Section two introduces the necessary background knowledge for better understanding. In section three we briefly overview the approach we devised. The process of constructing an ontology for the taxonomies is described in section four. Section five discusses how we built mappings among the ontologies. Section six introduces prototype tools. The last section describes the conclusion and future work.

2 Background

It is necessary to first make clear the definitions of the terms “taxonomy” and “ontology” because they are distinct concepts that we frequently misused as synonyms. A taxonomy is a particular classification arranged in a hierarchical structure organised by subtype-supertype relationships. Guarino defines that ”an ontology is a logical theory accounting for the intended meaning of a formal vocabulary [7].” Simply speaking, a taxonomy only contains the corresponding vocabulary with a hierarchical structure in the ontology; however an ontology can include axioms to define and restrict the semantics and relationships in the taxonomy, which is more important. Most ontologies have a taxonomy, but not all taxonomies can be easily converted to ontologies, due to ill-defined parent-child relationships. In a formal ontology, strict is-a interpretation is used for the taxonomy, i.e. every instance of a child is an instance of the parent. But in informal taxonomies parent-child relationships are often topic-oriented. Therefore an

important and difficult step in our project is first to convert all the taxonomies into corresponding ontologies.

One of the advantages of using ontologies is that they naturally support large-scale distributed information. When information resources commit to the same ontology then the same meaning is anticipated for any term from that ontology. Even when information resources commit to different ontologies, there are still methods to integrate the information [8], as long as the ontologies have certain relationships, e.g. their concepts are defined in terms of a common ontology or an alignment is provided. Some other advantages of ontologies for the integration of heterogeneous and distributed classifications in e-commerce are discussed in [6].

The basis of the Semantic Web is that there are a number of ontologies, and different information resources commit to the definitions in these ontologies [1]. The problem considered in this paper can be solved by using the Semantic Web approaches. There are multiple semantic web languages with different features that have been intensively researched and designed. OWL is the W3C recommendation for a web ontology language and is an extension of the Resource Description Framework (RDF). The OWL class constructors (see Table 1) and axioms can be used to express rich semantics. In this paper, we focus on OWL DL, the sublanguage of OWL that most closely corresponds to description logics (DL). Most DLs are decidable and have sound and complete reasoning algorithms. The worst-case complexity of SHOIN(D), the DL corresponding to OWL DL, is NEXPTIME-complete [16], but modern tableaux-based reasoners typically have very good performance.

Creating domain ontologies from industrial products categorization standards is not straightforward as it appears. We have seen the RDF-S version [10] and the DAML+OIL version [12] of UNSPSC. There also exists a prototype of a RDF representation of eCl@ss 4.1 based on an OWL Full ontology [2]. These ontologies, however, were created automatically and have not been used for later integration. They do not reflect the semantics of the underlying standards precisely and specifically enough and are thus not suitable for the basis of inference. Hepp [9] points out that many classifications are created from a procurement viewpoint, and this results in hierarchies that do not have the strict "is-a" semantics of rdfs:subClassOf. For example, eCl@ss typically includes parts and accessories as subcategories of each product in its hierarchy. He proposed to represent a concept in the classification using two concepts in the ontology, one is a generic concept and the other is for the taxonomy category (representing the concept plus related goods). The taxonomy concepts are arranged in a strict rdfs:subClassOf hierarchy corresponding to the classification. A new "annotation class" is created for each concept, and is made as rdfs:subClassOf both the generic and the taxonomy class. Any specific products are asserted to be of this type.

Ontology matching is actively researched. There are mainly three categories [4]. The first is ontology mapping between an integrated global ontology and local ontologies, and typical tools include LSD and MOMIS. The second is mapping

Table 1. OWL class constructors

Constructor	DL Syntax	Example
intersectionOf	C1 ⊓ C2	GasTurbine ⊓ AircraftPart
unionOf	C1 ⊔ C2	Door ⊔ Airframe ⊔ TailSection
complementOf	¬C	¬Aircraft
oneOf	{x1,...,x2}	{F15, F16}
allValuesFrom	∀P.C	∀partOf.Airframe
someValuesFrom	∃P.C	∃hasPart.Door
maxCardinality	≤ nP	≤10hasPart
minCardinality	≥ nP	≥2hasPart

between local ontologies which is more suitable for cases with many ontologies and typical tools include GLUE, MAFRA and ONION. The third allows a single coherent merged ontology to be created when ontologies which have the same or overlapping domains; and tools include PROMPT [13] and Chimaera [11]. Automated and semi-automated methods typically use syntactic techniques, linguistic resources and/or ontology structure to identify matches. To the best of our knowledge, no automated technique can discover alignments more complicated than subsumption between two named classes.

Omelayenko [14] surveys various integration approaches for ecommerce. Most of these techniques focus on syntactic matching [3]. The approach proposed by Corcho et al. [5] is most similar to ours. In their approach they separate e-commerce standards into generic ontologies, which provide coarse-grained classifications of products, and regional ontologies whose concepts share a common root by all mapping to the concepts in the generic ontologies. Below those two levels, there are catalogs and optional local ontologies. This mechanism makes classifications into a multi-layered structure. However, since Corcho et al. do not map generic ontologies to other generic ontologies, each regional ontology must map to each concept in any overlapping portions of these ontologies. Also, since they do not map regional ontologies to each other, it is not possible to align the most specific concept in these ontologies. Finally, we note that they only use equivalent, subclass and union-of axioms in their mappings, limiting the types of mappings that they can express.

Compared to that proposal, the agreed common ontology in our approach is created based on common vocabularies from all incorporated e-commerce standards. Additionally all standards are on the same layer so that we do not need preliminary work to determine their roles. The mapping based on this structure is clear and easily understood. This simple structure has significant flexibility for changes of ontologies since each ontology only needs to map to a single interchange ontology. The additional advantage is that the user of a standard does not need to understand all partners' standards in order to integrate with them, because that standard only needs to be integrated with the common ontology. Obviously this costs less both when initially building mappings among them and when new ontologies are incorporated.

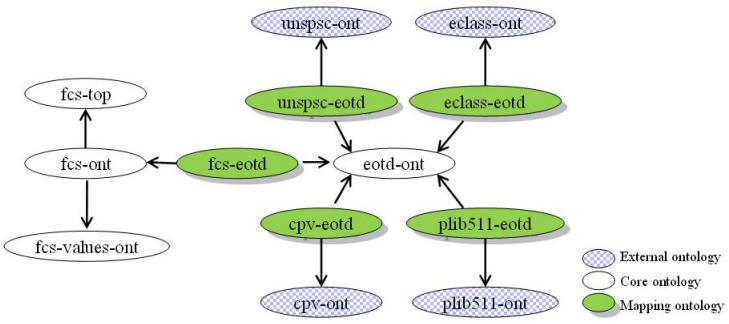

Fig. 1. The ontology network for the project. Each node is an ontology and the arrow points to another ontology which it extends from.

3 Approach

As introduced in the previous section, DL can be used to integrate taxonomies. To accomplish this, we must transform all taxonomies into OWL DL in order to describe concepts, terms and all expressive relationships in order to further embody higher level semantics. This step semi-automates the process of knowledge acquisition from the sources of information selected and adapts them to the ontological knowledge model. For all product ontologies, we assume that instances are "item of production," i.e. each instance is a specific part from a specific manufacturer. All created ontologies form an inter-related ontology network. Besides the FCS, the core of the network includes the eOTD which provides the means for comparing an external classification with the core FCS classification. It acts as a *lingua franca* whereby one classification can be mapped to another, which means classifications are not mapped directly to each other but by the eOTD intermediary. At the perimeter of the network (see Fig. 1) are all external ontologies derived from commercial taxonomies.

The next and critical step in our system is to formally define the terms in the FCS and external ontologies. We do this by writing axioms that ground the concepts defined in them with eOTD concepts. Therefore, the eOTD functions as the "standard" reference taxonomy against which other taxonomies are compared. Although these axioms could be included in the FCS and external ontologies themselves, for modularity reasons we place them in mapping ontologies. These mappings constitute "mediator" ontologies or navigational paths among different ontologies.

Assuming a sufficient intermediary ontology and axiomatization of the other ontologies, any concept can be translated to its subsuming concept in another ontology simply via logical entailment. We built a product classification translator that integrates all the taxonomies and analyzes potential reasonable relationships to search for "best available match" through multiple edges.

Using the translator's outcome we can solve a more practical requirement from the suppliers. The wrapper and compiler tools together fulfill this requirement

to translate product descriptions to the FCS classification scheme and can be applied to heterogeneous formats, like XML, spreadsheets and other semi-structured or structured documents.

4 Ontology Construction

The core ontologies include four ontology files for the FCS taxonomy and one for the eOTD. In addition, there are four other ontologies for external taxonomies: eCl@ss, CPV, UNSPSC and PLIB-511. When we created these four external ontologies, we tried to automate the processes and keep their original hierarchical structures, because this can prove that our method does not need significant work to change or adapt other classifications for our approach. The semantic gap between those classifications and the FCS is shortened by ontology mapping which will be described in section 5. Therefore in this section we mainly discuss the construction of core ontologies and the characteristics and difficulties of external taxonomies we found during ontology construction.

4.1 Naming Convention

In order to build mappings systematically in later steps, it is necessary to establish certain naming conventions. In generating ontology identifiers we took the original name from the taxonomy and removed all spaces and punctuations converting it to Camelback notation. In accord with RDF convention, we start classes with an upper case letter and properties with a lower case letter (e.g. "taperIncludedAngle"). Properties that have the same name but apply to different classes are distinguished by numbers at the end of the class name. If a child uses the same name to refer to a more specific concept, we append "All" in the name of its super class. If a class actually includes things that are not is-a children of the named concept, we append "Etc" because this indicates that the more general concept is improperly named. We use the underscore where a class name begins with a number, such as "_12".

Sometimes, this naming convention fails to adequately denote the concept, because punctuation may aid in interpreting a name. We use a rdfs:label to preserve the original name. Thus for example, the class name: StudAssembly-TurnlockFastener could be supplemented with the label: "STUD ASSEMBLY, TURNLOCK FASTENER" for a more natural English interpretation.

4.2 FCS Ontology

The FCS is an extensive taxonomy for naming, classifying, and describing items of supply under inventory control by the Department of Defense (DoD). The Defense Logistics Information Service (DLIS) is responsible for the FCS and uses the taxonomy to catalog and differentiate items of supply. Each item in the FCS is assigned a four-digit code by the government to designate various groups of common use. The first two digits is the Federal Supply Groups (FSG) code

identifying the group and the last two digits is the Federal Supply Class (FSC) code identifying the classes within each group. Each FSC has many Item Name Codes (INCs) which are five-digit numbers assigned by DLIS to each Approved Item Name (AIN). Although we don't discuss it in this paper, each INC also has many National Stock Numbers (NSNs) where each represents a different item of supply. The Master Requirement Code (MRC) is a four-character code assigned to a Federal Item Identification Guide (FIIG) requirement, and a set of MRCs can be used to indicate whether a requirement in a FIIG needs to be described for the item being identified.

Since the purpose of this project is to integrate external taxonomies with the FCS, our first step was to automatically create the FCS ontology from FCS database. The first ontology (fcs-top) for the FCS defines higher level terms in the FCS, such as the classes for FSGs and FSCs. Each FSC class is an rdfs:subClassOf its parent FSG class. This ontology contains all FSGs and FSCs. The second ontology (fcs-ont) for the FCS extends from fcs-top and defines classes for all the AINs in our scope. It includes properties for all mandatory MRCs of each AIN. Every AIN is an rdf:subClassOf its appropriate FSC class. The third ontology (fcs-values-ont) for FCS extends from fcs-ont and defines the values in the FIIGs of the FCS as instances in the ontology. To support the ontologies above, we created a meta ontology (fcs-meta) that defines annotation properties for recording the FSG, FSC, INC and MRC codes. This allows the ontology to be searched by DLIS personnel using their terminology.

Since this project was only a prototype, we focused on 128 AINs describing batteries, bearings, bushings, fasteners, gaskets and electronic circuits. In most cases, each AIN is represented in the ontology as a class and is identified by its INC in the fcs-ont. We use a minimum cardinality restriction to indicate properties defined as mandatory by the FCS. In principle, each item of production should have exactly one AIN, therefore we made all of the AINs under an FSC disjoint, and all FSCs under an FSG disjoint. In general disjointness axioms like these help in detecting errors in alignment axioms and can even be used to infer most specific relationships between classes in different ontologies. So we tried to make every disjoint relationship explicitly declared in every ontology we created. For instance, "fsc:BatteryNonrechargeable $\sqcap$ fsc:BatteryRechargeable $\equiv \bot$". Although we generally found the FCS to be a well-designed classification system, there were some defficiencies. For example, some different MRCs have the same syntactic words as name, e.g. MATT and MATL both have names "Material".

4.3 eOTD Ontology

The eOTD is an international standard for e-catalogs via the ISO 22745 designation used to create unambiguous language independent encoded descriptions of master data. It is designed to support industry classification by providing a classification neutral dictionary of names and attributes (or characteristics, properties) of items.

Since the eOTD is designed to be classification neutral, it does not contain a taxonomy. It only includes the most specific terms from each incorporated

classification scheme. One of these schemes is the FCS. The eOTD includes a class for each AIN and a property for each MRC. In creating our OWL ontology, we only included those classes and properties that are in the scope of the project. In section 5.1, we discuss how we enrich the eOTD with a class hierarchy to better support our mappings.

4.4 External Ontologies

In this sub-section, we briefly introduce the four representative external ontologies we selected in this project.

Ecl@ss is a German initiative to create a standard classification of material and services for information exchange between suppliers and their customers. We use the 6.0 version of eCl@ss which has over 25,000 nodes arranged in a four-level hierarchy. We only created classes for those concepts that were in our project scope. We did not have access to a machine-readable copy, the only source we accessed is the website (http://www.eclass-online.com/). The eCl@ss ontology has a limited set of properties.

UNSPSC is a coding system to classify both products and services for use throughout the global eCommerce marketplace and its partners include 3M, AOL, Arthur Andersen, BT, Castrol and others. There are over 20,000 nodes in the hierarchy, which like eCl@ss is four levels deep. A limitation of the UNSPSC is that it does not define any properties, and no definitions are given for classes. We wrote a simple program to translate an Excel spreadsheet describing the catalogue into RDF.

CPV is the European-wide classification system for public procurement contracts. It has over 8,000 nodes arranged in a hierarchy with 7 levels. It employs two different methods for building the taxonomy. The first is a straightforward hierarchy proceeding from more general concepts to specific instances. The second method is based on a type relationship model. Like UNSPSC, CPV also does not have any properties nor are concept definitions provided. As above, we automated data extraction from Excel spreadsheets.

PLIB [15] is the Parts Library series of international standards, and is defined by ISO 13584. It is developed and maintained by the ISO technical committee. PLIB is a data model. In this project we use PLIB-511, the PLIB fasteners dictionary as an external ontology. Our ontology includes all of the classes and properties from this dictionary. PLIB-511 has a finer granularity of classes, but the FCS has a superior breadth of coverage. It is the only external ontology considered that has a relatively large number of properties. PLIB specifies an electronic interchange format which facilitates automatic conversion to OWL.

Table 2 summarizes all the ontologies we created so far. Most of our classifications have limited or no properties, for instance eCl@ss, UNSPSC and CPV described above. However properties are important components for a full ontology. Some properties do exist in taxonomies though they are at a fairly high level and lack detail. Furthermore, there are no definitions, at least in UNSPSC, CPV and the online version of eCl@ss that we were able to access over the Web. At best, there are lists of alternatives for specified terms. Also sometimes the

modeling is inconsistent and fails to always observe implicit modeling conventions. These conditions introduce difficulties when mapping ontologies.

Table 2. Statistics of ontologies created

Ontology	Classes	Properties	Depth	Mean Siblings
fcs-top	735	365	2	14
fcs-ont	128	2	3	13
eOTD	194	180	5	10
eCl@ss	313	18	4	8
UNSPSC	228	0	4	13
CPV	208	0	7	5
PLIB-511	186	204	6	12

5 Ontology Mapping

This section discusses how to build mappings between ontologies for each corresponding taxonomy. All the original ontologies and mapping ontologies form an inter-connected ontology network (recall Fig. 1).

Our mapping construction is done iteratively as described in Fig. 2. As a preliminary step in the mapping process, we enriched the eOTD vocabulary to provide a solid basis for later steps. In the mapping process, we found and built a number of mappings manually, then used a reasoner to evaluate the validity of these constructs. If there were some errors in our previous manual construction, additional editing or correction of these mappings were needed. Otherwise, we repeated this process iteratively based on all of the preceding knowledge.

5.1 Enriching the eOTD

Given the wide variety of terms, attributes and properties across the taxonomies, we need to find a way to accommodate all of the differences. As we mentioned in section 4.3, one of the purposes of the eOTD is to serve as the common vocabulary that can be used to define concepts in all of other ontologies. However, this mapping is complicated by the fact that the eOTD does not have a hierarchy. Therefore we found it helpful to add more general concepts to the ontology. Not only are those concepts convenient when we are defining terms from more abstract external ontologies, but they enable definitions even when the eOTD terms provide incomplete coverage of the varieties of a product. We refer to these concepts as "abstract classes," and populate them by analyzing the FCS and, to a lesser extent, the external ontologies. Here we use two guidelines. The first is to remove one or more modifiers from class names. For instance, we created two abstract classes "Bearing" and "BearingRoller" for the AIN "Bearing, Roller, Self-Aligning." Another guideline is to identify "foundational" classes from FSGs and FSCs. For instance, we create four abstract classes "BearingsAntifriction," "BearingsPlain," "BearingsMounted" and "BearingsUnmounted" since we have

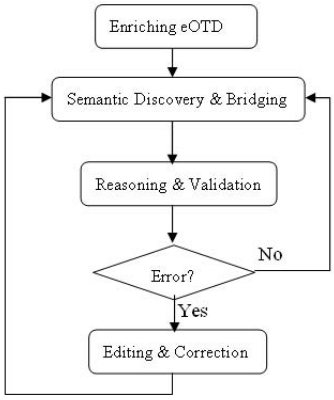

Fig. 2. The process of building mappings

FSCs "Bearings, Antifriction, Unmounted," "Bearings, Plain, Unmounted" and "Bearings, Mounted." The classes in the eOTD serve a similar role to Hepp's generic classes [9]. However, as we discuss in the next section, we provide explicit formal mappings to the original classes from commercial classification schemes.

5.2 Semantic Discovery and Bridging

Ontology mapping is the process of aligning the classes and properties of two ontologies. We do this by creating axioms in a mapping ontology that state equivalence, subclass, superclass relationships, exception, restriction, etc. In order to establish mappings between two ontologies, it is necessary to discover their logical and semantic relationships. Discovery of these relationships is the precondition to building the mappings.

We note that automated ontology alignment approaches [4] are not very helpful here. The differences between classification schemes are not as simple as the equivalence or subclass relations between named classes typically found by such systems. Even terms with identical names are often semantically different. For this reason, our alignment was a manual process that involved looking at names and ontology structures, and when possible we used definitions and actual product descriptions.

Our discovery focuses on two kinds of mapping axioms: owl:equivalentClass and rdf:subClassOf. In all mapping axioms, the left-hand side is a description composed of classes from one ontology, and the right-hand side is a description composed of classes from another ontology. Although equivalence is the most desirable matching, it is difficult to find it. Concepts that appear at one level in one classification frequently appear at different levels in the other taxonomies. When equivalence matches cannot be found, we tried to specify a most specific subsumer and most general subsumee. For example,

cpv:PrimaryBatteries $\sqsubseteq$ eOTD:BatteryAssemblyAll
eOTD:BatteryThermal $\sqsubseteq$ cpv:PrimaryBatteries.

To build these mappings, we used the most appropriate OWL constructs (see Table 1 for details) to express their relationship, such as those listed below.

1. Union (A $\equiv$ B $\sqcup$ C)

Sometimes a term name is the literal aggregation of two or more other terms from different classification. From the conventional English point of view, the phrase "A and B" usually does not mean the logical conjunction of A with B, but instead the disjunction of the concepts, e.g. "fsc:KnobsAndPointers $\equiv$ eOTD:Knob $\sqcup$ eOTD:Pointer."

2. Intersection (A $\equiv$ B $\sqcap$ C)

When a concept's name describes multiple independent characteristics, it is best defined as an intersection of classes representing these characteristics. For instance, "fsc:BearingAntifrictionUnmounted $\equiv$ eOTD:BearingAntifriction $\sqcap$ eOTD:BearingUnmounted."

3. Exclusion (A $\equiv$ B $\sqcap \neg$C)

We often found that the different classifications have similar concepts with significant overlap, but there are often exceptions. In this case, we can correlate them by pointing out the difference between them. Exclusion can be used to specify that A and B are the same except for some C. This idiom is especially useful when trying to align a procurement-based taxonomy with formal "is-a" taxonomies. For instance, "eOTD:BearingPlain $\equiv$ eCl@ss:PlainBearing $\sqcap \neg$ eCl@ss:PlainBearingParts."

4. Class vs. property distinction (A $\sqsubseteq \exists$ P.{a, b, c})

In some cases, the distinction made by a class in one ontology is made by a property in another. We found this to be true when mapping PLIB-511 to the FCS/eOTD, because it is a much more specialized taxonomy. We can use someValuesFrom or hasValue axioms to account for this. For example, "PLIB:HexagonHeadTappingScrewWithAFlatEnd" is classified partly according to two properties head style and point style. Meanwhile we can enumerate the values of both properties. So this item should have one of these values on the property. Thus "PLIB:HexagonHeadTappingScrewWithAFlatEnd $\sqsubseteq$ $\exists$eOTD:headStyle.{eOTD:Hexagon}" and "PLIB:HexagonHeadTappingScrewWithAFlatEnd $\sqsubseteq$ $\exists$eOTD:pointStyle.{eOTD:Flat, eOTD:Flat2, eOTD:Flat3, eOTD:Flat4}." This restriction demonstrates that this concept in PLIB-511 has a determined value for the property head style, and simultaneously has a limited set of values on another property, point style.

5.3 Reasoning and Validation

Although domain experts and ontology developers can try to build all knowledge into an ontology, it is possible that they did not declare all assumptions or supplied incorrect axioms. Thus after each iteration of semantic discovery and bridging, a consistency check for the mapping of the ontology is necessary. A typical example of an inconsistency is when a class is a subclass of two disjoint

classes. The most likely cause of this is an incorrect mapping axiom, but it is also possible that the disjointness condition is invalid.

Besides checking consistency, the second purpose of reasoning is to compute subsumption for the ontology. Subsumption determines which classes are necessarily subclasses of other classes based on the classes' descriptions and our bridging axioms. If insufficient subsumptions are discovered, then we need to add new mapping axioms or make the existing ones more specific.

6 Implementation

In previous sections, we have introduced the one-to-one ontology mappings. Now we will illustrate how they can be expanded to the mappings across the whole ontology network. The knowledge of domain experts is often limited to a few familiar classification schemes, and perhaps even only specific classes of product within these schemes. It is unrealistic to expect one person to know how to map to all other classification schemes. But if two classifications of different domains both have a correspondence (mapping) to a common third domain or more generally there is a correspondence path of multiple bridges between them, we can incorporate a DL reasoner to automatically infer a virtual correspondence between these two classifications. In other words, this method can implicitly "merge" two classifications via other ontologies.

Based on this idea we implemented a translator tool that intended to selects the most specific FCS concepts that include the terms appearing in the four external classification systems. It takes in a selected external ontology, core ontologies and the bridging ontologies between them together as input, and queries the DL reasoner about every kind of possible logical relationship we are interested in. Finally we visualize the quality of these mappings on the interface. We use HAWK[1], a homegrown Java Semantic Web API, to parse the ontologies and use a DIG interface to load these ontologies into FaCT++, although any DIG-compliant reasoner could be used.

We query the reasoner about equivalent and direct super class of each item in a selected external ontology. In the result we filter out those classes which are not in FCS ontologies, because we are only interested in those classes. Obviously, equivalent classes are preferred, because they imply an exact translation. Failing this, the most specific subsuming class is sufficient because this is an accurate (although more general) description of the original class. If the reasoner does not return any direct superclasses that are from the FCS, we then recursively query the classes that are returned until we find an FCS class. We call these results "indirect" subsumption. So the translator finally output three kinds of relations: equivalence, direct and indirect subsumption.

Table 3 gives a brief summary of translation results. We note that generic concepts rarely have matches due to classifications grouping things from different perspectives. As a result, very specialized schemes like PLIB-511 have a high degree of matches with FCS, while generic schemes like CPV have fewer matches.

[1] http://swat.cse.lehigh.edu/downloads/index.html#hawk

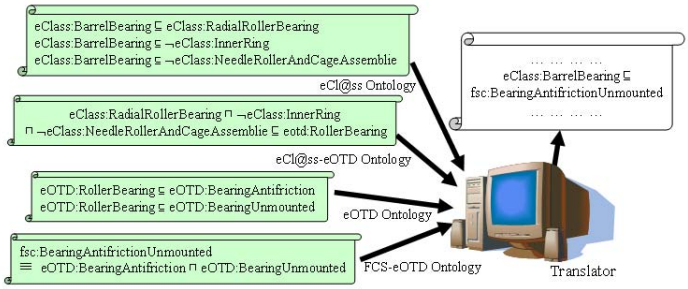

Fig. 3. An example of translation

For this reason, we recommend that suppliers use the most specific classification schemes available.

Table 3. Summary of computed matches. The scope means the number of concepts where we build mappings. The general scope includes all concepts in the scope and all concepts descended from them. The matching percentage is the sum of all three kinds of matches divided by the general scope.

Ontology	Scope	General Scope	Equivalence	Direct Subsumption	Indirect Subsumption	Matching Percentage
eCl@ss	86	191	13	21	78	58.64%
UNSPSC	79	103	7	55	18	77.67%
CPV	43	117	1	8	23	27.35%
PLIB-511	86	86	0	13	72	98.83%

Fig. 3 is an example of translation. A cataloger needs to register a new supplier who has used eCl@ss to describe their products, including a Barrel Bearing (code: 23.05.09.12). In order to determine how these items can be classified in the FCS, the cataloger can use our product classifciation translator. Given the eCl@ss ontology, its mapping to the eOTD ontology, and the mapping of the eOTD to the FCS ontology, the translator can determine that "eCl@ss:BarrelBearing $\sqsubseteq$ fsc:BearingAntifrictionUnmounted" is entailed. Thus the Barrel Bearing product should be clasified under this FSC.

To help both suppliers and customers to easily control and understand these translations, we developed a graphical user interface (see Fig. 4). Through it, users can select external classifications being mapped into the FCS classification. The results include the concept names and codes of both sides and the relationship mentioned above. The results of class and properties are separated in two pages. They are all listed in a tabular form which can be sorted by any column. Thus users can clearly read whether these mappings are satisfied.

In our prototype, the ontologies are of moderate size (typically hundreds of classes and a few thousand axioms) and as a result the translation is completed in tens of seconds including the reasoner's computation. However, in production

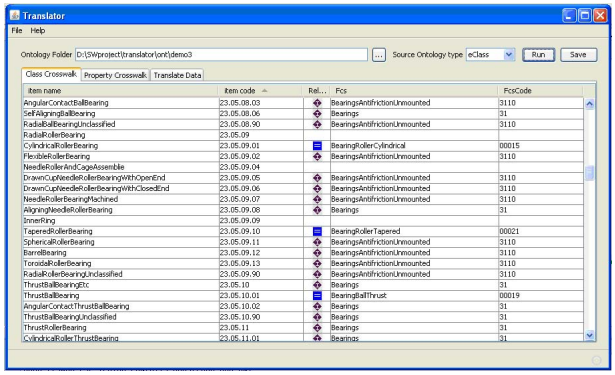

Fig. 4. The graphical user interface of the translator

the ontologies will be much larger. In order to save time, we provide a way to compile these mappings into persistent storage form, since these ontologies do not change very frequently (usually at the scale of months). This can make applications that use the translations much more efficient. The compiler can be rerun any time when source or target ontologies change. Therefore we can compare the mappings of different versions and more importantly subsequent steps can directly utilize these existing mappings rather than translating the ontologies "on the fly."

In order to translate data items from a target product description into FCS terms, we need to make the technical data items queryable by the OWL ontology language. So we provide a wrapper, i.e. a structure, for these data items to be translated into RDF Format. Via this process, the source data instances are transformed into standard form which uses terminology from the ontology version of the target classification.

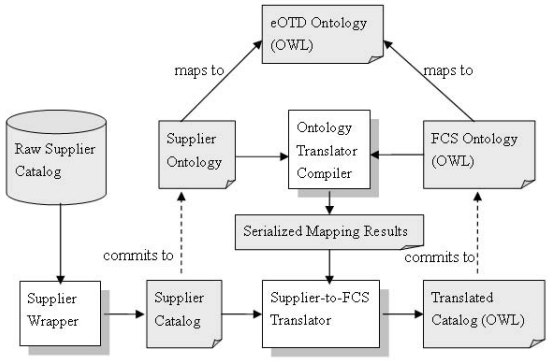

Fig. 5. The complete process of compilation. The white boxes are our prototype tools

As Fig. 5 shows, the wrapper takes in the supplier's raw catalog data which could be in the format of plain text, webpages or XML documents, and rewrites it into an RDF form that commits to the appropriate external ontology. Then the previously compiled translation results are incorporated to translate the wrapped catalog into a new catalog which commits to the FCS ontology. Specifically, the system first determines the category and type information of a catalog product description. After that it uses the previously compiled mapping relation (mentioned above) between this concept and the target concept in the FCS to rewrite the description of this product. With this tool, suppliers can easily find each product's corresponding descriptions in the terms of the FCS classification system.

7 Conclusion and Future Work

In this paper, we have demonstrated how Semantic Web technology can be used to ease integration of various e-commerce classification schemes. With some extensions, the eOTD provide an excellent common vocabulary and OWL DL is expressive enough to relate classification schemes that have different levels of specificity and different assumptions about what a parent-child relationship means. Ideally, we would like to see suppliers and maintainers of standard classification schemes create their own mappings to the eOTD, or have them use a standard product ontology that is so mapped, but in the meantime it is possible for these mappings to be developed by a third party. This would reduce the cost of integrating such ontology with the FCS or any other terminology. The clear advantage is less ambiguous classifications and the ability to automatically interchange with other schemes.

Future work includes extending the scope to include other kinds of products and improving our tools. We would like to refine the translator so that translating a source class to a property value works correctly and properties are translated in a more comprehensive way. We also believe that there are only a finite number of ways of modeling any given product, and would like to investigate the possibility of using automated methods to align new ontologies with the most similar ontology in our ontology network, achieving an economy of scale as the ontology network grows.

Acknowledgments

This work is supported by the Defense Supply Center Philadelphia, Philadelphia PA, and the Defense Logistics Agency, Ft. Belvoir, VA under contract SP4701-07-C-0006. We thank Peter Benson of the Electronic Commerce Code Management Association (ECCMA), Gerald Radack of Concurrent Technologies Corporation, Xinlei Wu and Ameet Chitnis for their help with the project.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284, 34–43 (2001)
2. Bizer, C., Wolk, J.: RDF Version of the eClass 4.1 Product Classification Schema (retrieved June 2009), http://www.wiwiss.fu-berlin.de/suhl/bizer/ecommerce/eClass-4.1.rdf
3. Bowers, S., Delcambre, L.: Representing and Transforming Model-Based Information. In: Proceedings of the Workshop on the Semantic Web at ECDL 2000, Lisbon, Portugal, September 21 (2000)
4. Choi, N., Song, I.-Y., Han, H.: A Survey on Ontology Mapping Sigmod Record. ACM SIGMOD Record archive 35(3) (2006)
5. Corcho, O., Gomez-Perez, A.: Solving Integration Problems of Ecommerce Standards and Initiatives through Ontological Mappings. In: IJCAI 2001 Workshop on Ontologies and Information Sharing, pp. 131–140 (2001)
6. Fensel, D., McGuinness, D.L., Schulten, E., Ng, W.K., Lim, E.-P., Yan, G.: Ontologies and Electronic Commerce. IEEE Intelligent Systems 16(1), 8–14 (2001)
7. Guarino, N.: Formal Ontology in Information Systems. In: Proceedings of FOIS 1998, Trento, Italy, pp. 3–15. IOS Press, Amsterdam (1998)
8. Heflin, J., Hendler, J.: Dynamic Ontologies on the Web. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI 2000), pp. 443–449. AAAI/MIT Press (2000)
9. Hepp, M.: Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards. Int'l Journal on Semantic Web & Information Systems (IJSWIS) 2(1), 72–99 (2006)
10. Klein, M.: DAML+OIL and RDF Schema representation of UNSPSC (retrieved June 4, 2009), http://www.cs.vu.nl/~mcaklein/unspsc/
11. McGuinness, D., Fikes, R., Rice, J., Wilder, S.: An Environment for Merging and Testing Large Ontologies. In: Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR 2000), Breckenridge, Colorado, pp. 12–15 (April 2000)
12. McGuinness, D.L.: UNSPSC Ontology in DAML+OIL (retrieved from June 4, 2009) http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml
13. Noy, N., Musen, M.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: Proceedings of the AAAI 2000 Conference, Austin, TX (2000)
14. Omelayenko, B.: Syntactic-Level Ontology Integration Rules for E-Commerce. In: FLAIRS Conference 2001, pp. 324–328 (2001)
15. Pierra, G.: Context-explication in conceptual ontologies: The PLIB approach. In: Proceedings of 10th ISPE International Conference on Concurrent Engineering: Research and Applications (ce 2003): Special Track on Data Integration in Engineering, pp. 243–254 (2003)
16. Tobies, S.: The complexity of reasoning with cardinality restrictions and nominals in expressive Description Logics. J. of Artificial Intelligence Research (JAIR) 12, 199–217 (2000)