PCS2OWL: A Generic Approach for Deriving Web Ontologies from Product Classification Systems

Alex Stolz[1], Bene Rodriguez-Castro[2], Andreas Radinger[1], and Martin Hepp[1]

[1] Universitaet der Bundeswehr Munich, D-85579 Neubiberg, Germany
{alex.stolz,andreas.radinger,martin.hepp}@ebusiness-unibw.org
[2] Technical University Munich, D-81675 Munich, Germany
bene.rodriguez@tum.de

Abstract. The classification of products and services enables reliable and efficient electronic exchanges of product data across organizations. Many companies classify products (a) according to generic or industry-specific product classification standards, or (b) by using proprietary category systems. Such classification systems often contain thousands of product classes that are updated over time. This implies a large quantity of useful product category information for e-commerce applications on the Web of Data. Thus, instead of building up product ontologies from scratch, which is costly, tedious, error-prone, and high-maintenance, it is generally easier to derive them from existing classifications. In this paper, we (1) describe a generic, semi-automated method for deriving OWL ontologies from product classification standards and proprietary category systems. Moreover, we (2) show that our approach generates logically and semantically correct vocabularies, and (3) present the practical benefit of our approach. The resulting product ontologies are compatible with the GoodRelations vocabulary for e-commerce and with schema.org and can be used to enrich product and offer descriptions on the Semantic Web with granular product type information from existing data sources.

Keywords: #eswc2014Stolz

1 Introduction

The classification of products and services plays a crucial role for many businesses and business applications [1]. It enables reliable and efficient electronic transactions on product data across organizations in a dynamic domain, characterized by innovation and a high degree of product specificity. Product classes generally allow for intelligent decision-making and operations over aggregated data. More specifically, the ability to operate on groups of products is generally superior to applying heuristics on unstructured product descriptions, especially at tasks for generalizing or discerning products. For instance, a search for a personal computer relying on textual matches not only returns personal computers but likely also related accessories or books that discuss the topic *personal*

computers. Class membership information helps to reliably distinguish between personal computers and related, but not necessarily relevant, products. Moreover, it adds a mechanism to query for all existing personal computers, which otherwise, with heuristics, is difficult and expensive.

In practice, organizations often arrange products and services according to product classification systems. At the same time, the number of quality, practically relevant product ontologies on the Web is still limited [2], because most ontology engineering work is done in the context of academic research projects where efforts rarely go beyond *toy* status. Thus, a cost-efficient solution able to accommodate business needs on the Web of Data would be greatly appreciated.

Product classification systems are suitable candidates for creating high-quality and low-cost product ontologies for the Web [3]. In many fields of e-commerce for example, where a domain is typically composed of thousands of classes and properties, it is difficult to engineer domain ontologies manually, because that would imply to get hold of a large number of concepts. Moreover, the conceptual dynamics [2] underlying the domain of products and services, determined by a continuous innovation progress and the high degree of specificity, make the manual creation of product ontologies even more problematic. Let us exemplify the situation by comparing the release sizes [4] of different versions of eCl@ss [5], a comprehensive industry standard for the classification and description of products and services: eCl@ss 5.1.4 had defined 30,329 classes in 2007, whereas eCl@ss 6.1, only announced two years later, was already counting 32,795 classes. The changes become even more evident for eCl@ss 6.1 and eCl@ss 8.0 BASIC with an increase of 20%, reaching 39,041 concepts within only three years. Thus, instead of engineering new ontologies, it is often more practical to derive product ontologies from works already in place, i.e. to reuse existing industrial taxonomies, as argued in [3]. This has several benefits: (1) the product classifications provide a comprehensive coverage of the conceptual domains, and that often in multiple languages; (2) there is no significant overhead involved for maintaining derived product ontologies; on the contrary, they are automatically kept up-to-date with amendments to the classifications conducted by domain experts in response to changes in the real-world; (3) existing industrial standards are popular and thus already in wide use to classify product instance data. In other words, a large amount of products in relational databases are already classified according to product categorization standards. Also numerous Web shops create and maintain proprietary category systems together with their product catalogs. Hence, instead of manually crafting complex domain ontologies and thereby in a sense reinventing the wheel, it is often sensible to unlock the potential of existing, well-maintained hierarchical structures and classify products on the Semantic Web according to them.

In this paper, we present a generic approach and a fully-fledged, modular, and largely automated tool for deriving Web ontologies from product classification systems. We show that our approach generates logically and semantically correct ontologies that (1) establish canonical URIs for every conceptual element in the original schema; (2) preserve the taxonomic structure of the original classifica-

tion while making its categories usable in multiple contexts; (3) comply with the GoodRelations vocabulary for e-commerce [6] and schema.org; and (4) can be readily deployed on the Web of Data. The results of our transformation unlocks additional semantics that enable novel Web applications. Thanks to the enrichment of product master data and a more granular description of offers by virtue of product ontologies, search engines and other consumers of structured data, can take advantage of product type information for product search, comparison and matchmaking.

2 Product Classification Systems

For the scope of this research, we distinguish two groups of classification schemes relevant to the domain of commercial products and services. These are *product classification standards* and *proprietary product category systems* (or structures). We use the broader term *product classification system* (or PCS for short) to refer to any artifact from any of the two groups. The main aspects of both groups are discussed in this section. Additionally, there is further relevant information that cannot be included here due to space limitations, but is available online[3]. This supplementary material gathers a series of key attributes for every classification system comprising version, organization(s) authoring and managing the classification, available data sources, official report, target usage domain, intended regional use, and level of multilingual support.

2.1 Product Classification Standards

Product classification standards (or product categorization standards) are widely accepted knowledge structures often consisting of thousands of categories. They typically comprise: (a) hierarchical structures for the aggregation of products, which allow for example spend analysis or reasoning over hierarchical relations; (b) common features and values related to product categories; and (c) multilingual descriptions of the elements that conform the standard.

The product classification standards that we considered at the time of this research are: Classification of Products by Activity (CPA) [7], Central Product Classification (CPC) [8], Common Procurement Vocabulary (CPV) [9], eCl@ss [5], *ElektroTechnisches InformationsModell*[4] (ETIM) [10], FreeClass [11], Global Product Classification (GPC) [12], proficl@ss [13], and *Klassifikation der Wirtschaftszweige*[5] (WZ) [14]. The featured standards are grounded on industry consensus and exist for various business fields, be it horizontal or vertical industries. eCl@ss, proficl@ss, and GPC, for example, describe a wide range of products from multiple industrial sectors. By contrast, CPV is intended for the procurement domain, whereas ETIM is focused on the field of electronics. Two standards, CPA and WZ, put forward classifications of comprehensive economic

[3] http://www.ebusiness-unibw.org/ontologies/pcs2owl/
[4] Engl.: ElectroTechnical Information Model
[5] Engl.: Classification of Economic Activities

activities instead of products *per se*. Nonetheless, commercial products can be classified against them and their use is common among governmental publishers of statistical data. To solve potential ambiguity problems of product names, standards such as eCl@ss, ETIM, and proficl@ss, include synonyms to provide discriminatory features [15] and to retain higher recall in product search scenarios. Furthermore, many standards (CPA, CPV, FreeClass, and WZ) contain translations into various languages.

2.2 Proprietary Product Category Systems

Proprietary product category systems (or catalog group systems, category structures) are also suited for organizing products and services. Unlike product classification standards, catalog group systems are generally characterized by less community agreement. Single organizations or small interest groups instead of communities or standardization bodies are taking the lead for the development of such category structures. Thus, they are accepted only by a relatively small number of stakeholders, and their usage is limited to a narrow context, e.g. to represent a navigation structure in a Web shop. Some examples of catalog group hierarchies considered in the context of this paper are proprietary product taxonomies like the Google product taxonomy [16] and the productpilot category system [17] (the proprietary category structure of a subsidiary of Messe Frankfurt), as well as product categories transmitted via catalog exchange formats like BMEcat[6] [18]. The latter can take advantage of both product categorization standards and catalog group structures in order to organize types of products and services and to contribute additional granularity in terms of semantic descriptions [19].

3 Deriving Product Ontologies from Hierarchical Systems

In this section, we present a generic, semi-automated approach to turn standards and proprietary product classification systems (PCS) into respective product ontologies. Subsequently we outline the conceptual architecture of our proposal, followed by a description of the conceptual transformation.

3.1 Conceptual Architecture

Fig. 1 depicts the conceptual approach of PCS2OWL[7]. The tool consists of a modular architecture that builds upon three layers, namely *parser*, *transformation* process, and *serializer*. It only requires a moderate amount of initial human labor, mainly to prepare the import modules (parsers) for the respective classification systems, as indicated by the dashed rectangle in Fig. 1. This task includes

[6] Developed by *Bundesverband Materialwirtschaft, Einkauf und Logistik (BME)*, Engl.: Federal Association of Materials Management, Purchasing and Logistics.

[7] Short for "product classification systems to OWL", available online at http://wiki.goodrelations-vocabulary.org/Tools/PCS2OWL

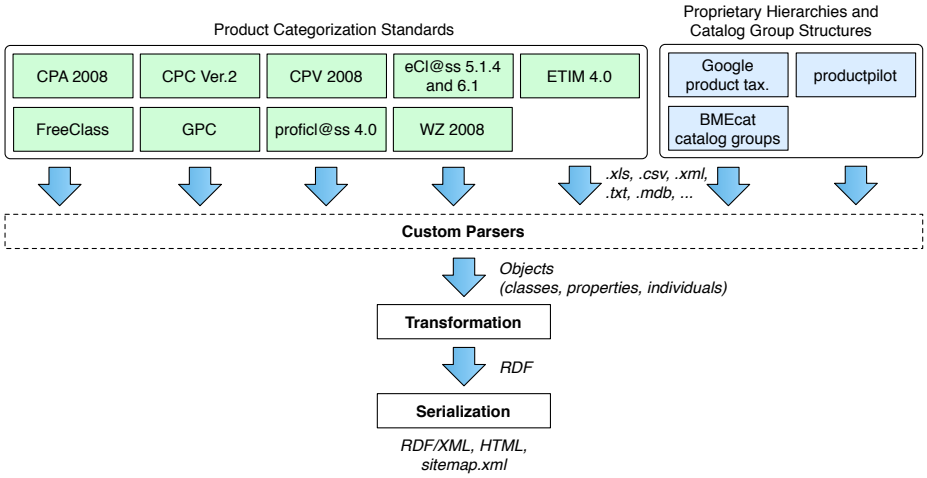

Fig. 1. Conceptual architecture of PCS2OWL

the logic for mapping the taxonomy and setting up the discerning capabilities of property types. The parsers' purpose is to load categories, features, and values of product classification systems into an internal model, which specifies ontology classes, properties, and individuals. The transformation and serialization processes are then fully automated. In the transformation step, the internal model, consisting of entities for classes, properties, and individuals, is turned into an RDF model that describes the final ontology. At this stage, also the logical rules from the parsers are applied to the internal model. Finally, the RDF model is serialized as RDF/XML, and all other files required for the on-line deployment of the product ontologies are created accordingly.

In the context of this paper, we developed custom parsers for a number of popular categorization standards and proprietary taxonomies for products and services, previously introduced in Section 2 and outlined in Fig. 1. Since the parsers have to be hand-crafted, the input formats of the source files of the classification systems do not matter much. For our conversions e.g., we had to deal with Excel spreadsheets (.xls), comma-separated value files (.csv), extensible markup language files (.xml), database tables (.mdb), and plain text files (.txt).

The effort required to develop a parser module is negligible compared to hand-crafting a product ontology from scratch. For simple classification systems with only classes and no properties such as GPC or Google, we extended the empty parser template with only twenty lines of custom code. Even the most complex parser module that we have created so far (FreeClass) required less than 200 lines of code, including sophisticated rules for raising the data quality of the resulting product ontology.

3.2 Transformation of a Product Classification System

A core aspect of the transformation step is the creation of the classes in the resulting ontology based on the source PCS being processed. To create the ontology classes, the PCS2OWL tool relies on the GenTax approach introduced in [20], whereby it is possible to generate a consistent OWL ontology while preserving the taxonomic structure of the original categories in the PCS. In order to do so, the GenTax method creates *two* OWL classes in the target ontology from each category in the PCS. The first is a broader taxonomic class that represents the category from the PCS in the target ontology. The second is a context-specific class, in our case in the domain of products and services. For a given category on the original PCS identified as *"ID"*, let us refer to the *pair* of OWL classes that GenTax creates as *C_ID-gen* and *C_ID-tax*, following the naming convention of the original GenTax specification [20].

There are additional design decisions that are applied in the conversion process to create the classes and the class structure of the resulting ontology: (1) all *C_ID-tax* taxonomic classes are arranged in a subsumption class hierarchy via the *rdfs:subClassOf* relation to preserve the hierarchical structure of the corresponding categories in the original PCS; (2) every *C_ID-gen* context-specific class is defined as a subclass of *gr:ProductOrService* of the GoodRelations ontology [6] via the *rdfs:subClassOf* property to state that it represents all instances of a certain product or service in the real world; (3) every *C_ID-gen* context-specific class is at the same moment also a subclass of the corresponding *C_ID-tax* taxonomic class, to preserve its traceability to the category in the original PCS that it was derived from; and (4) no subsumption relations exist between *C_ID-gen* context-specific classes given that as a class of an actual product or service, it is not possible to know in an automated fashion whether a subsumption relation between two *C_ID-gen* classes is applicable and valid in the real world.

Fig. 2 illustrates an example that results from the conversion of the following fragment of the English version of the Google product taxonomy [16]:

```
Cameras & Optics > Cameras > Digital Cameras
Cameras & Optics > Cameras > Disposable Cameras
```

Fig. 2 exhibits all *four* design decisions of the GenTax algorithm outlined previously. The right side shows the taxonomic class hierarchy, whereas the left part describes the context-specific class hierarchy. The black solid arrows stand for the *rdfs:subClassOf* relation. As indicated, (1) the taxonomic classes represent the categories in the Google taxonomy and preserve the same hierarchical structure; (2) the context-specific classes represent actual products and services and hence, are subsumed by *gr:ProductOrService*; (3) all context-specific classes are at the same time subclasses of their respective taxonomic class, e.g. the context-specific class *C_Cameras-gen* is a subclass of the taxonomic class *C_Cameras-tax*; and (4) no subsumption relation is imposed upfront between the context-specific classes, thus in visual terms they are arranged as mutual pair-wise siblings.

The adoption of the GenTax approach provides several features to the resulting ontologies produced by the PCS2OWL tool. GenTax creates meaningful,

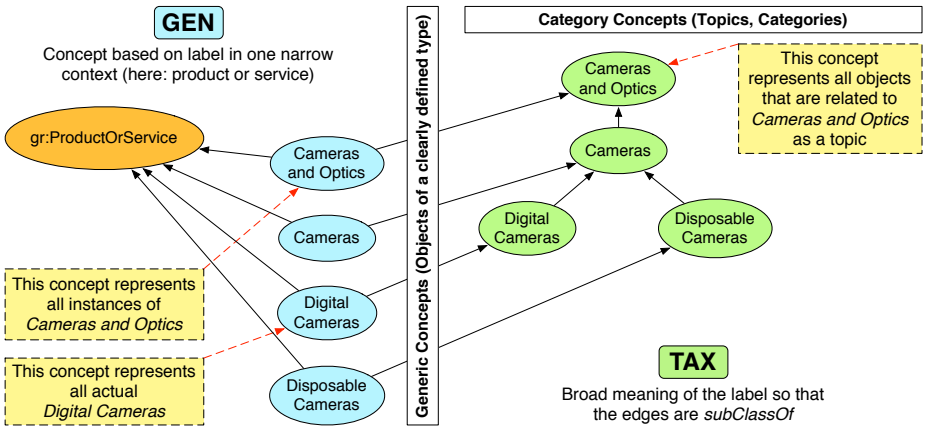

Fig. 2. GenTax applied on an extract of Google product taxonomy (cf. [20])

practically useful product classes (i.e. *"-gen"* classes on the left side of Fig. 2) by defining these as subclasses of *gr:ProductOrService*, which at the same time, renders the resulting ontology compatible with GoodRelations and schema.org. By preserving the hierarchical structure of the PCS (i.e. *"-tax"* classes on the right side of Fig. 2), GenTax allows the execution of generalization/specialization queries based on the original PCS. For example, searching for the common category *C_Cameras-tax* in order to get the union of all instances of the classes *C_DigitalCameras-gen* and *C_DisposableCameras-gen*. The use of the *rdfs:subClassOf* relation in the taxonomic classes, means that no reasoning capabilities beyond the widely supported RDFS reasoning are required to *navigate* through the taxonomic structure of the original PCS in the generated ontology. Additionally, for traceability and provenance purposes, every class indicates the ontology that describes it by taking advantage of the *rdfs:isDefinedBy* property; and moreover, every taxonomic class specifies a hierarchy code annotation property (*:hierarchyCode*) to link it to the corresponding category code used in the source classification system.

3.3 Converting Property Types and Related Values

In addition to the extraction of OWL classes from hierarchical classifications, PCS2OWL converts features and feature values of PCS, thus contributing additional semantics to categories. The different types of properties that are supported by the tool are in line with the GoodRelations ontology and consist of qualitative properties (*gr:qualitativeProductOrServiceProperty*), quantitative properties (*gr:quantitativeProductOrServiceProperty*), and datatype properties (*gr:datatypeProductOrServiceProperty*). Similarly, our tool distinguishes two enumeration types, namely qualitative values (*gr:QualitativeValue*) and quantitative values (*gr:QuantitativeValue* of type *xsd:float* or *xsd:integer*, e.g. values that indicate ranges), plus literal values with datatypes (*xsd:float*, *xsd:integer*, *xsd:boolean*, or *xsd:string*).

Custom rules and heuristics guide the distinction of the property types and related values. They have to be provided as part of the parser modules in order they can be applied in the subsequent transformation step where respective OWL properties are generated automatically. Thus, the quality of the conversion strongly depends on the correctness of these logics: As a general rule of thumb, a numerical value accompanied by a unit code in the classification system yields a quantitative value in the resulting product ontology, and not a qualitative value or a datatype literal. Some classification standards even make the intended type of features and values explicit, e.g. ETIM indicates logical values with an "L" metadata flag, hence best mapped as boolean literals in RDF.

3.4 Serialization and Deployment

In this section, we describe the serialization and deployment of the resulting product ontologies. This includes deciding on a canonical URI pattern for publishing the entities on the Web, and providing alternative ways to support standards-compliant Web ontology deployments.

The product classes and related entities in the ontologies obey a common URI pattern, which is comprised of (1) the base URI of the ontology; (2) a prefix to help humans distinguish URIs of different entity types, namely *C_* for classes, *P_* for properties, and *V_* for values; (3) an identifier unique in the context of the category system, that for categories is typically the hierarchy code; and, for classes, (4) a suffix to distinguish generic (*-gen*) from taxonomic (*-tax*) classes. Following this pattern, the URI of a context-specific class "Disposable Cameras" (hierarchy code *10001488*) in the GPC product ontology is

```
http://www.ebusiness-unibw.org/ontologies/pcs2owl/gpc/C_10001488-gen
```

PCS2OWL offers two deployment alternatives for product ontologies, namely based on *hash* and *slash* URIs. The first option generates a single comprehensive dump of the RDF graph, which is serialized as RDF/XML. The downside of this approach is the huge file size aspect that can make it infeasible for large classification systems. By contrast, the *slash*-based option generates a series of small RDF files, comprising separate files for all taxonomic and generic classes, and, if available, also for properties and individuals. This has the advantage that it allows serving smaller chunks of code for individual elements compared to its full dump counterpart. Moreover, with this option the tool creates a navigable documentation consisting of a set of interlinked HTML pages that mimic the subsumption hierarchy. The two deployment alternatives imply different URI patterns, that are

```
http://example.org/pcs#C_1234-gen -> hash-based
http://example.org/pcs/C_1234-gen -> slash-based
```

Besides the creation of RDF/XML and HTML files, PCS2OWL generates a *Semantic Sitemap*, and an *.htaccess* file for the easy deployment on an Apache Web server. Content negotiation is ensured using best practice patterns described online[8]. For *slash* URIs it means that by dereferencing an arbitrary entity URI

[8] http://www.w3.org/TR/swbp-vocab-pub/#recipe5

(e.g. a class URI), an HTML-preferring client is redirected to a respective HTML document using the HTTP response status code *303 See Other*. Similarly, the client retrieves RDF/XML, if the media type supplied with the HTTP *Accept*-header is `application/rdf+xml`. In this sense, our approach constitutes a full LOD-compliant deployment [21].

4 Evaluation

In the evaluation we focus on two key aspects, namely on the correctness of the conversion results, and on the amount of new product classes, properties, and enumerations obtained that are readily available for the Web.

4.1 Correctness of the Derived Product Ontologies

In this part of the evaluation, we were interested in whether the product ontologies correctly reflect the elements and the hierarchical structure provided by the product classification systems. We first did a quantitative comparison of the conceptual elements in the product classification systems and all classes, properties and individuals of the corresponding product ontologies. For that purpose we examined the number of concepts in the source files or database tables and the number of files produced for related types of concepts, e.g. the number of taxonomic classes in ontologies. If the numbers matched, it implied that the concepts were properly reflected in the product ontologies, which actually was the case for all of the ontologies that we built.

We complemented and further confirmed our previous findings by an experiment conducted on a product ontology derived from the Google product taxonomy [16]. The taxonomy file is available online[9] as plain text. It is line-based and characterized by a category tree which hierarchical structure is expressed using delimiting angle brackets as follows:

```
Food, Beverages & Tobacco > Beverages > Coffee > Coffee Pods
```

The taxonomy is read from the left starting with the most generic concept and getting more specific moving to the right. Accordingly, *Coffee* is a more specific concept than *Beverages* with respect to Google's product taxonomy. Our idea was basically to reverse-engineer the original taxonomy starting from the product ontology that we loaded into a SPARQL endpoint. A set of appropriate SPARQL queries permitted us to build up the whole hierarchy in a *top concept* → ... → *bottom concept* fashion. We then concatenated the respective RDFS labels using the exact same delimiters as advocated by the Google product taxonomy file format. And finally, the results of the concatenation were compared to the lines in the original source file. This way we were able to recreate an equivalent copy of the original file, which confirms the validity of our conversion. The single steps of our evaluation approach are described online[10] in more detail.

[9] http://www.google.com/basepages/producttype/taxonomy.en-US.txt
[10] http://www.ebusiness-unibw.org/ontologies/pcs2owl/evaluation/

Table 1. Statistics of product classification standards and category systems

Classification system	Number of					Class distr. (%)
	levels	classes	properties	individuals	top-level c.	
CPC Ver.2	5	4,409	0	0	10	18
CPA 2008	6	5,429	0	0	21	53
CPV 2008	4	10,419	0	0	254	6
eCl@ss 5.1.4	4	30,329	7,136	4,720	25	18
eCl@ss 6.1	4	32,795	9,910	7,531	27	16
ETIM 4.0	2	2,213	6,346	7,001	54	8
FreeClass 2012	4	2,838	174	1,423	11	21
GPC 2012	4	3,831	1,710	9,562	37	17
proficl@ss 4.0	≤ 6	4,617	4,243	6,815	17	36
WZ 2008	5	1,835	0	0	21	33
Google prod. tax.	≤ 7	5,508	0	0	21	17
productpilot	≤ 8	7,970	0	0	20	28
BMEcat	na	na	0	0	na	na

4.2 Statistics on New Product Classes and Properties

In Section 1, we have argued that our approach produces a large number of readily usable product classes for the Web that to craft and maintain manually is impracticable. In order to support this claim, we report in the current section relevant statistics about the derived product ontologies[11].

As a preliminary step, we loaded all product ontologies into a SPARQL endpoint. Storing each product ontology as a different named graph (`urn:cpa`, `urn:gpc`, etc.) allowed us later to execute SPARQL queries based on their graph names. To give an example, we used the SPARQL 1.1 query of Listing 1.1 (prefix declarations omitted) to determine the number of hierarchy levels in the product ontologies. We executed the query repeatedly where in every step we incremented the property path length by one unit until we obtained no more results.

```
SELECT (COUNT(DISTINCT ?c) AS ?num_classes) WHERE {
  GRAPH <urn:gpc> {
    ?c a owl:Class .
    ?c rdfs:subClassOf{3} ?sc .
    FILTER NOT EXISTS {?c rdfs:subClassOf gr:ProductOrService}
  }
}
```

Listing 1.1. Calculating the number of hierarchy levels of PCS

Increasing the property path length from 3 to 4 in the provided example returns zero results, meaning that the hierarchy depth of the product ontology is four, i.e. the longest existing path consists of four classes linked by three consecutive *rdfs:subClassOf*-relationships. The FILTER statement of the query assures that only taxonomic classes are regarded, excluding those classes defined as products or services which would lead to otherwise incorrect results.

[11] http://www.ebusiness-unibw.org/ontologies/pcs2owl/

As reported in Section 2, our research took into account ten popular product classification standards, among them two different versions of eCl@ss, and three proprietary category structures. The common abbreviations of the PCS together with the versions that have been converted are given in the first column of Table 1. The upper part lists the numbers for the product categorization standards, whereas the lower three rows of the table represent the proprietary category systems. For BMEcat we cannot report specific numbers, since the standard permits to transmit catalog group structures of various sizes and types. Columns two to six capture the number of hierarchy levels, product classes, properties, value instances, and top-level classes for each product ontology. It is worth noting that some of the product ontologies have a fixed number of hierarchy levels (e.g. eCl@ss has four levels), while for others the numbers vary (e.g. proficl@ss, which has up to six levels). Similarly, some of them are quite shallow (e.g. ETIM with 2 levels), while others provide deep hierarchies (e.g. CPA with 6 levels) with sometimes redundant concept names at consecutive levels. The large quantity of entities (classes, properties, individuals) implies an extensive coverage of the product or services domain, which, if built up manually, would be prohibitively expensive and time-consuming. Besides product classes, some product ontologies also contain properties and individuals that contribute valuable product details for the Semantic Web. Lastly, the seventh column indicates the distribution of classes within the derived product ontology (cf. Table 2 in [22]). This distribution is measured as the percentage of classes that belong to the largest top-level class with respect to the total number of classes in the ontology. This value describes the topology of the hierarchical structure and is thus an indicator for the quality of the product ontology. For example, in CPA one ("manufactured products") of the 21 top-level classes contains more than half of all the classes in the standard, while the classes in ETIM are more evenly distributed across various branches (only 8% of all classes belong to the largest class "hand tools").

Among the classification systems with multilingual support, CPA is the one with the most translations featuring class labels in 26 languages on average. Other product ontologies that also support multiple languages are CPV with an average of 22.9 languages, FreeClass with 6.9, WZ and the productpilot category system with both 2. The variety of languages supported increases the chance of finding products annotated with product classes more easily on the Web.

5 Discussion

This section presents a series of e-commerce use case examples that embody some of the novel opportunities that search engines and other consumers of structured data can exploit in areas such as product search, comparison, and matchmaking. These opportunities arise from using the now available Web product ontologies from PCS2OWL that allow to articulate more granular product descriptions across both the Web of Documents and the Web of Data.

Let us consider e.g., an online retailer interested in improving its product trading and data management processes. One enhancement consists in the adop-

tion of the GPC classification standard instead of developing a custom scheme from scratch, leveraging the GPC Web ontology. Our retailer has published on the Web a snippet in Microdata syntax as in Listing 1.2, describing a specific disposable camera. For readability, the qualified names of the vocabulary URIs involved are used. They rely on the prefix declaration of *gr:* for GoodRelations [6], *gpc:* for the GPC product ontology[12], and *s:* for schema.org[13].

```
<div itemtype="http://schema.org/SomeProducts" itemid="#p1234" itemscope>
  <link itemprop="additionalType" href="http://www.ebusiness-unibw.org/
      ontologies/pcs2owl/gpc/C_10001488-gen" />
  <meta itemprop="name" content="Kodak 35mm Single Use Camera Flash" />
  <!-- additional features -->
</div>
```

Listing 1.2. Annotation example in Microdata syntax

Classification of Product Descriptions. Listing 1.2 specifies a disposable camera *p1234*, that is defined as an instance of the class *s:SomeProducts* (equivalent to *gr:SomeItems*) and identified by a fragment in the scope of the Web document URI. Thanks to the *additionalType* property in schema.org Microdata, *p1234* is an instance of the class *gpc:C_10001488-gen* as well. This definition, together with the existing linkage across the classes *gpc:C_10001488-gen*, *gpc:C_10001488-tax*, and the property *gpc:hierarchyCode* in the GPC Web ontology, materializes the product *p1234* on the Web as an instance of the category *10001488* labeled as "Disposable Cameras" in the original GPC classification standard.

Navigation over Product Data. The adoption of the GPC Web ontology would allow our online retailer to *navigate* along the product categories of the original GPC standard. Applied to the example in Listing 1.2, this navigation path is determined by the super- and subclasses of *gpc:C_10001488-tax*, which are defined via the *rdfs:subClassOf*-relationship. For example, the immediate parent class of *gpc:C_10001488-tax* (the category of our camera) is *gpc:C_68020100-tax*[14]. Or, in terms of the original schema, the GPC product category *68020100* "Photography" is the parent category of *10001488* "Disposable Cameras".

Web Data Format Descriptions of Product Data. The fact that product classes are published on the Web using URIs renders them applicable for use with common Web data formats, such as Microdata, RDF in attributes (RDFa), and Facebook Open Graph (OGP). Product annotations in those syntaxes can also lead to improvements on the current state of the document-based Web, namely in the form of search-engine result snippets (known as "rich snippets") and other mid-term benefits that may arise from providing more semantics.

6 Related Work

This paper partially builds upon previous works in the area of transforming classification standards into Web ontologies. The challenges in the conversion

[12] http://www.ebusiness-unibw.org/ontologies/pcs2owl/gpc/
[13] http://schema.org/
[14] http://www.ebusiness-unibw.org/ontologies/pcs2owl/gpc/C_68020100-tax

of product classification standards were already discussed in [23,3], whose findings led towards the development of the GenTax algorithm in [20], still a core component of PCS2OWL. The subsequent initial release of the GoodRelations ontology [6] motivated the first transformation of the eCl@ss standard (5.1.4)[15] relying on the GenTax methodology as a GoodRelations compliant ontology.

Alternatively, there have been previous efforts to convert other product classification schemes also supported by PCS2OWL: Most notably CPV ([24], another effort[16]), primarily used to streamline the procurement and tendering process in the public sector. On a broader scope, the research in [25] provides the most recent and comprehensive survey of methods and tools for the refactoring of most types of non-ontological resources (NORs) into ontological resources (ORs), i.e. Web ontologies. A comprehensive qualitative framework is put forward to categorize NORs based on their characteristics. One of the types of NORs acknowledged in the work are actually the general classification schemes for any given domain, such as those considered in this paper for products. In fact, two methods [26], again GenTax, and a tool, SKOS2OWL[17], are identified to focus on the conversion of classification scheme NORs specifically into Web ontologies.

Yet, in summary, to the best of our knowledge, PCS2OWL remains as the only methodology readily supplied with tool support, that extends the features and capabilities of all the conversion efforts previously mentioned, on at least one, if not several of the following fronts: (1) the level of automation; (2) modular architecture supporting the conversion of an arbitrary number of classification systems; (3) the application to a broad set of non-ontological resources, i.e. almost all relevant classification schemas; (4) traceability including preservation of the taxonomic structure between the elements in the original classification scheme and those in the derived Web ontology; (5) improved support for properties and enumerations; (6) high degree of configuration options aimed at deployment on the Web of Linked Open Data (LOD); and, lastly, (7) compliance to the GoodRelations and schema.org ontologies, which currently allows for the publishing on various Web data formats.

7 Conclusions

The ontology engineering task in the domain of products and services is typically tedious, costly, and time-consuming. To master this problem, we presented a generic method and a toolset for deriving product ontologies from existing product classification standards and proprietary category systems in a semi-automatic way, which is usually superior to building them up manually in several aspects. For example, it successfully addresses the generally large number of concepts in product categorization standards and the conceptual dynamics inherent to the domain of products and services. We have supported our contribution by converting 13 product classification systems of different scopes, sizes,

[15] http://www.heppnetz.de/projects/eclassowl/
[16] http://linked.opendata.cz/resource/dataset/cpv-2008
[17] http://www.heppnetz.de/projects/skos2owl/

and structures, and have shown that we can generate practically relevant product ontologies while effectively preserving the original taxonomic relationships. These ontologies are ready for deployment on the Web of Linked Open Data. Furthermore, we exemplified how products can be annotated using the derived product ontologies, rendering them more visible and discernible on the Web. In particular, employing product classes to semantically annotate product instances empowers product data consumers to find and aggregate products and respective offers with less effort. For example, they could be readily used for assisting faceted search over semantic e-commerce data.

As future work, we are planning to extend the set of available parsers by additional product classification systems, and to publish already converted product ontologies which, at the time of writing this paper, we were not yet granted permission due to lack of copyright clearance. Moreover, we think that our product ontologies could attract related research fields, such as finding correspondences across product classification systems by means of ontology matching. Similarly, we should point out that our generic toolset could be easily adapted to convert classification systems even outside the product domain.

Acknowledgments. The work on this paper has been supported by the German Federal Ministry of Education and Research (BMBF) by a grant under the KMU Innovativ program as part of the Intelligent Match project (FKZ 01IS10022B), and by the Eurostars program (EU 7th Framework Program) of the European Commission in the context of the OPDM project (FKZ 01QE1113D).

References

1. Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M., Flett, A.: Product Data Integration in B2B E-Commerce. IEEE Intelligent Systems **16**(4) (2001) 54–59
2. Hepp, M.: Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies. IEEE Internet Computing **11**(1) (2007) 90–96
3. Hepp, M.: Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards. International Journal on Semantic Web and Information Systems (IJSWIS) **2**(1) (2006) 72–99
4. eCl@ss: Category:Products - wiki.eclass.eu. http://wiki.eclass.eu/wiki/Category:Products#Release_Sizes
5. eCl@ss: eCl@ss Classification and Product Description. http://www.eclass.de/
6. Hepp, M.: GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In Gangemi, A., Euzenat, J., eds.: Knowledge Engineering: Practice and Patterns. Volume 5268 of LNCS. Springer (2008) 329–346
7. European Commission: Regulation (EC) No 451/2008 of the European Parliament and of the Council of 23 April 2008 establishing a new statistical classification of products by activity (CPA) and repealing Council Regulation (EEC) No 3696/93. Official Journal of the European Union **L145/51** (June 2008)
8. United Nations Statistics Division: The Central Product Classification (CPC) Version 2.0. http://unstats.un.org/unsd/cr/registry/cpc-2.asp

9. European Commission: Commission Regulation (EC) No 213/2008 of 28 November 2007. Official Journal of the European Union **L074/51** (March 2008)
10. ETIM Deutschland: ETIM 4.0 – Das Klassifizierungsmodell der Elektrobranche. http://www.etim.de/
11. Handle, O.: Konzeption und Realisierung eines branchenübergreifenden Produktklassifikationssystems für das Bauwesen unter Nutzung der produktspezifischen Fachkompetenz der Baustoffindustrie. Master's thesis, MCI Management Center Innsbruck (2007)
12. GS1: Global Product Classification (GPC): The Global Language for Classifying Goods. 3rd edn. GS1 (April 2005)
13. proficl@ss International: proficl@ss - der Branchenstandard. http://www.proficlass.de/
14. Statistisches Bundesamt: Klassifikation der Wirtschaftszweige (WZ 2008). Wiesbaden: Statistisches Bundesamt (2008)
15. Navigli, R.: Word Sense Disambiguation: A Survey. ACM Comput. Surv. **41**(2) (2009) 10:1–10:69
16. Google Merchant Center: The Google product taxonomy. http://support.google.com/merchants/bin/answer.py?hl=en&answer=1705911
17. Messe Frankfurt: productpilot. http://www.productpilot.com/
18. Schmitz, V., Leukel, J., Kelkar, O.: Specification BMEcat 2005. Bundesverband Materialwirtschaft, Einkauf und Logistik e. V. (2005)
19. Stolz, A., Rodriguez-Castro, B., Hepp, M.: Using BMEcat Catalogs as a Lever for Product Master Data on the Semantic Web. In Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S., eds.: The Semantic Web: Semantics and Big Data. Volume 7882 of LNCS. Springer (2013) 623–638
20. Hepp, M., De Bruijn, J.: GenTax: A Generic Methodology for Deriving OWL and RDF-S Ontologies from Hierarchical Classifications, Thesauri, and Inconsistent Taxonomies. In Franconi, E., Kifer, M., May, W., eds.: The Semantic Web: Research and Applications. Volume 4519 of LNCS. Springer (2007) 129–144
21. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. 1st edn. Morgan & Claypool (2011)
22. Hepp, M., Leukel, J., Schmitz, V.: A Quantitative Analysis of Product Categorization Standards: Content, Coverage, and Maintenance of eCl@ss, UNSPSC, eOTD, and the RosettaNet Technical Dictionary. Knowledge and Information Systems **13**(1) (2007) 77–114
23. Hepp, M.: A Methodology for Deriving OWL Ontologies from Products and Services Categorization Standards. In: Proceedings of the 13th European Conference on Information Systems (ECIS 2005), Regensburg, Germany (2005) 1–12
24. Polo Paredes, L., Álvarez Rodríguez, J.M., Azcona, E.R.: Promoting Government Controlled Vocabularies for the Semantic Web: the EUROVOC Thesaurus and the CPV Product Classification System. In: Proceedings of the Semantic Interoperability in the European Digital Library workshop (SIEDL 2008), co-located with 5th European Semantic Web Conference, Tenerife, Spain (2008) 111–122
25. Villazón Terrazas, B.: A Method for Reusing and Re-engineering Non-ontological Resources for Building Ontologies. PhD thesis, Univ. Politécnica de Madrid (2011)
26. Hakkarainen, S., Hella, L., Strasunskas, D., Tuxen, S.: A Semantic Transformation Approach for ISO 15926. In: Proceedings of the 1st International Workshop on Ontologizing Industrial Standards (OIS 2006), co-located with Advances in Conceptual Modeling - Theory and Practice, Tucson, AZ, USA (2006) 281–290