

ProdLight: A Lightweight Ontology for Product Description Based on Datatype Properties

Martin Hepp

Digital Enterprise Research Institute (DERI), University of Innsbruck

mhepp@computer.org

Abstract: Web pages representing offerings of products and services are a major source of data for Semantic Web-based e-commerce. This data could be useful for numerous applications, e.g. (1) more precise product search engines and shopping bots, (2) aggregation or enrichment of multi-vendor catalogs using public product descriptions, or (3) the automated discovery of additional alternatives based on the combination of multiple items. While there are already some ontologies for products and services available, they are very large in size (20 – 70,000 classes), and thus not always suitable as ontology imports. In this paper, we take a different approach: We represent the semantics of offerings on the Web using a very lightweight ontology of datatype properties in combination with popular classifications like UNSPSC and eCl@ss. We then demonstrate how this representation can be mapped easily to comprehensive ontologies for products and services like eClassOWL¹. Our approach provides a straightforward solution for annotating offerings on the Web while avoiding the overhead of importing fully-fledged products and services ontologies in any single annotation. We can show that our proposal has technical advantages and eliminates legal problems when reusing existing standards.

1. Introduction

Web pages representing offerings of products and services are a major source of data for Semantic Web-based e-commerce. This data covers technical and commercial aspects and could be useful for numerous future applications. Firstly, it could be used by novel product search engines and shopping bots that identify suitable alternatives for a given need and a given set of preferences. Secondly, the data could be used for assembling, augmenting, or maintaining multi-vendor catalogues for e-shops. For example, product descriptions and technical features from pages published by the manufacturers of a product could be exploited for maintaining the product descriptions in a

¹ <http://www.heppnetz.de/eClassOWL/>

retail e-shop. Thirdly, it could support the discovery of additional alternatives based on the combination of multiple items.

However, most of the data is available only in a format suitable for rendering by browsers and interpretation by humans; and despite much research in using ontologies for supporting such scenarios, no comprehensive and practically useful framework exists so far for making Web offerings accessible at a semantic level.

In B2B scenarios, quite comprehensive classification standards for products and services have evolved, namely UNSPSC [1] and eCl@ss [2]. Their reuse for e-commerce applications of the Semantic Web vision is however non-trivial, as has been shown in [3], [4], and [5].

With eClassOWL, there exists now a consistent OWL variant of the eCl@ss standard, containing ontology classes for more than 25,000 different goods categories plus more than 5,000 properties for representing product characteristics [6]². While this ontology is freely available and can be readily used for many applications, the idea of annotating existing Web resources that describe products and services still faces several problems.

a) Size

Due to their original design purpose of aggregating data on an enterprise-wide level, most available classifications are very large in size with usually more than 20,000 categories.

b) Intellectual Property Rights

It has recently been stressed that many standards are subject to copyrights and other types of intellectual property rights [7]. Thus, creating and publishing an ontology by transforming an existing standard usually requires explicit agreements with the owners of the standard [8]. In the case of eCl@ss, such an agreement has been established. For UNSPSC, however, the issue has not yet been settled.

c) Unclear semantics of categories in standard classifications

Most informal classifications lack a clear notion of what it means to be an instance of a respective category. This is no problem when using a classification schema in a well-defined context known to all consumers of the data, but it causes some problems when deriving ontologies from such classifications, since the latter requires a clear semantics. For example, UNSPSC codes are used both for classifying expenses and for describing

² A similar ontology derived from UNSPSC called unspscOWL has been created but is waiting for copyright clearance.

actual products. Deriving an ontology from a classification thus includes important modeling decisions in this issue [4]. As one direction, there has been work on adding a formal semantics to informal classifications by [9, 10].

From a practical perspective, we need to be able to describe (1) actual products and services *instances*, (2) *models* of products and services, and (3) *entities just related to a certain type of goods* (e.g. invoices).

d) Type of relationship between a product category and a business entity

When representing the products and services domain, a lot of semantics is kept in the relation between a Web resource and a type of product or service. In fact, in most cases we do not just want to say that a specific Web resource is an instance of a specific product category. Much more, we want to represent a different type of relationship between a Web resource and a product definition, e.g. “This Web page contains an offer of product instances that meet the following specification” or “The company identified by this URI repairs products of the following kind”. A valuable standard providing common concepts for this problem is the UNSPSC Business Function Identifier (BFI) [11]. It is a simple two-digit value that reflects the kind of relationship between a business entity and a product category. Table 1 shows the currently supported values.

Table 1. UNSPSC Business Function Identifiers [11]

UNSPSC BFI	Meaning
10	Rental or Lease
11	Maintenance or Repair
12	Manufacturer
13	Wholesale
14	Retail
15	Recycle
16	Installation
17	Engineered
18	Outsource

In short, we assume that the creation and usage of true ontologies derived from products and services classifications is in general useful, for it provides a proper conceptual model for describing offerings, demands, and related items like invoices properly. However, it may not be realistic to annotate a majority of current Web shops by references to such ontologies in the near future, mainly for practical reasons. In particular, the scalability of current reasoners is still a bottleneck, which implies that drawing inferences about hundreds of thousands of offerings instances may not yet scale. Also, the amount of reasoning supported by such ontologies, e.g. eClassOWL, is rather limited. This is because popular product classifications have just four hierarchical levels with a limited amount of branching. Table 2 shows the median of branching between the levels of hierarchy in eCl@ss 5.1 and UNSPSC

7.0901. In here, the median tells us the maximum number of descendents for the lower half of the categories. We can see that from the third level to the fourth, half of all categories have not more than two descendents. This level of branching is particularly interesting, because the fourth level contains classes for actual commodities. UNSPSC is a bit more advanced in the branching at the lowest level with up to six descendents for the lower half of all categories. Still, one may ask whether it always makes sense to import a very large ontology for any operations on product data if the amount of reasoning is that limited.

Table 2. Amount of branching in popular product classification standards [data taken from 12].

	Median	
	eCI@ss 5.1	UNSPSC 7,0901
Top level → 2 nd level	18	5
2 nd level → 3 rd level	6	4
3 rd level → 4 th level	2	6

As a consequence, we are proposing a very lightweight mechanism that allows capturing the product semantics of Web offerings by referring to popular classification standards. This mechanism should avoid the burden of replicating the whole standard in the form of a huge ontology but still be upward compatible to such comprehensive ontologies. It should also be possible to work with respective annotations in a meaningful way even in the absence of scalable reasoning, e.g. on the level of plain RDF.

1.1. Related Work

The potential of using ontologies for e-commerce scenarios has been stressed e.g. by [13], [14], [15], and [16]. Gupta and Qasem argued that Semantic e-commerce may reduce the price dispersion in markets. [17]. In the information systems community, the benefits of product content standards have been discussed e.g. by [18].

A major focus of previous works on using ontologies in e-commerce scenarios was on the *integration* of product data, i.e. the challenge of harmonizing catalogue items or expenses referring to incompatible classifications. The complexity of this problem has been described e.g. in [19] and [20]. Respective tasks in B2B relationships have been detailed by [21]. The future role of ontologies on B2B marketplaces has been discussed in [22]. A prototype of catalogue data integration based on ontologies and ontological mappings was reported in [23]. An approach of identifying equivalences and other types of mappings between multiple product schemas is presented in [24].

Modeling aspects of product representation and respective ontology engineering tasks are covered by [25], [26], and [27].

The reuse of existing standards for e-commerce scenarios has been suggested e.g. in [28] and [29]. A domain-specific analysis of the problems faced when reusing product classifications for constructing products and services ontologies is given in [4]. A more generic analysis of the transformation of classifications into ontologies is presented in [10]. Recently, the W3C has proposed SKOS [30] as a very lightweight schema for representing classifications in the Semantic Web. While SKOS variants from standards can be created almost fully automatically, the resulting formalization cannot be directly used as a product and services ontology like eClassOWL, while its size will still be very large.

Currently, there exist at least the three ontologies for products and services ontologies, which were derived from eCl@ss or UNSPSC [31], [32], and [6]. While the last one is very current and also includes product attributes, the two former ones are somewhat dated snapshots of past UNSPSC versions. A comparison is given in [4].

1.2. Our Contribution

In this paper, we (1) describe how the semantics of offerings on the Web can be represented using a very lightweight ontology in combination with popular classifications like UNSPSC and eCl@ss. We then show how this representation can be mapped easily to comprehensive ontologies for products and services like eClassOWL. Our approach provides a straightforward solution for annotating offerings on the Web while avoiding the overhead of importing fully-fledged products and services ontologies in any single annotation.

2. The ProdLight Ontology

In the following, we describe a novel approach for using the class identifiers of common product and service classification schemas for the annotation of Web offerings. The main idea is to capture the class identifiers as literal values of simple datatype properties in the form “hasClassificationCode 1234” instead of making an offering an instance or a subclass of a fully-fledged products and services ontology like eClassOWL. This approach

requires only a very small ontology. At the same time, it is easy to create mappings to eClassOWL at any time.

This requires the following steps:

1. We define four classes
 - a. `ProductOrServiceOrRelatedInstance`
 - b. `ProductOrService`
 - c. `ProductOrServiceProxyInstance`
 - d. `ProductOrServiceRelatedEntity`
 and make b, c, and d subclasses of class a.

The class `ProductOrServiceOrRelatedInstance` contains everything that is either an actual product or service, an entity that acts as a proxy for multiple actual product or services instances (which themselves are not exposed, i.e. unknown), or items that are related to product or services categories but which are no actual products or services (e.g. related invoices)

The class `ProductOrService` is a subclass of the former class and contains only actual product or services instances, i.e. individuals on which property rights can be established and transferred.

The class `ProductOrServiceProxyInstance` is a work-around for such products and services instances that are existentially quantified but not directly exposed. This is necessary, because in many situations, Web shops do not publish actual instances of a product or service, but just state that such exist. Since existential quantification, which would be the proper modeling for such situations, is computationally costly and makes a respective OWL ontology fall into OWL DL, we use instances of this class to represent existentially quantified product offerings.

The class `ProductOrServiceRelatedEntity` contains everything that may be characterized by a product or services category but is not a product or service instance (or “instance proxy”) itself.

In OWL Abstract Syntax, the respective class definitions are as follows:

```

Class (ex:ProductOrServiceOrRelatedInstance partial)
Class (ex:ProductOrService partial
  ex:ProductOrServiceOrRelatedInstance)
Class (ex:ProductOrServiceProxyInstance partial
  ex:ProductOrServiceOrRelatedInstance)
Class (ex:ProductOrServiceRelatedEntity partial
  ex:ProductOrServiceOrRelatedInstance)
  
```

2. For the categories themselves, we just define one single owl:DatatypeProperty, e.g. `relevantEclassCategory_v5.1` with the range of a string, and use *the primary keys of categories in eCl@ss version 5.1 as the literal value*

In OWL Abstract Syntax, the respective property definition will be as follows:

```
DatatypeProperty(ex:relevantEclassCategory_v5.1
    range(xsd:string))
```

This can happen in parallel for multiple categorization schemes or versions from the same schema. For instance, we can define an additional datatype property for UNSPSC version 8:

```
DatatypeProperty(ex:relevantUnspscCategory_v8
    range(xsd:string))
```

If we allow the parallel usage of multiple classifications (e.g. UNSPSC and eCl@ss) or multiple values from the same standard, we need to be clear about the semantics of such assertions. The natural interpretation is that a resource annotated using two such statements reflects the *intersection* of both categories. For instance, if we use `relevantEclassCategory_v5.1` with the value “AAB29200202” (“Photo Cameras”) plus `relevantUnspscCategory_v9` with the value “45121504” (“Digital Cameras”), then the actual meaning of this shall be that the resource is related to the intersection of both; in this case, that it is related to *digital* cameras.

This aspect may be obvious for logicians; however, domain experts must be advised that a resource that contains two different offerings (e.g. cell phones and cell phone antennas) must be treated as two different resources (e.g. by adding URI fragment identifiers `#phone` and `#antenna`). Otherwise, a domain expert may add the UNSPSC codes for both categories to the same resource, which would be equivalent to the intersection of cell phones and cell phone antennas and thus, most likely, an empty set.

In addition to capturing the type of good, there is a need of expressing the *type of business function* that is supported for a particular product or service category, e.g. whether a Web page offers the sale, repair, or disposal of a particular type of goods. A first approach for capturing this aspect is by defining an additional datatype property that reflects the business function in the form of UNSPSC Business Function Identifier value (see [11] for details).

In OWL Abstract Syntax, the respective property definition will be as follows:

```
DatatypeProperty(ex:supportedBusinessFunctionsUNSPSC-BFI
    range(xsd:int))
```

In here, we also need to agree upon the semantics of multiple property assertions for one resource. We suggest defining the semantics of this property such that multiple statements imply that the *union* of the respective business functions is supported. In our opinion, this is intuitive; it also means that we can express support for multiple business functions (“We sell and lease out boats”) without duplicating the resource descriptions.

Instead of using `xsd:int` for the UNSPSC Business Function Identifier, we could also create instances for any of the types as specified in Table 1. However, since we are aiming at the most lightweight solution, we decided against that. We are in parallel working on a fully-fledged ontology for business functions, into which the proposed lightweight approach can easily be grounded later. This can be achieved using similar patterns as for mapping categories to eClassOWL classes as shown in section 4.

3. Example

In the following, we give a brief example of our approach in plain OWL DLP. Even though we are using `owl:class` for upward compatibility reasons, no reasoner support is necessary at this stage.

We assume there are four entities:

1. `CameraInvoice`: An invoice related to a digital camera purchase.
2. `CameraWebsite`: A Web site on which cameras are *sold to end users (retail)*.
3. `myCamera`: An actual camera.
4. `CameraModelSony123`: A Sony camera make and model, of which multiple actual instances exist, which will all share several technical features, e.g. weight, pixel resolution, etc. – however, the actual instances are only existentially quantified.

In the example, “AAB29200202” is the identifier of the category “Photo Camera” in eCl@ss 5.1. In UNSPSC v8, the closest category “Digital Cameras” has the identifier “45121504”[1]. The UNSPSC Business Function Identifier “14” means “Retail” [11].

Using the approach described in section 4.3.1, this leads to the following statements in OWL Abstract Syntax:

```
Individual(ex:CameraInvoice
  type(ex:ProductOrServiceRelatedEntity)
  value(ex:relevantEclassCategory_v5.1
    "AAB29200202"^^<http://www.w3.org/2001/XMLSchema#string>
  )
)
```

```
Individual(ex:CameraModelSony123
  type(ex:ProductOrServiceProxyInstance)
  value(ex:relevantEclassCategory_v5.1
    "AAB29200202"^^<http://www.w3.org/2001/XMLSchema#string>
  )
)
```

```
Individual(ex:CameraWebSite
  type(ex:ProductOrServiceOrRelatedInstance)
  value(ex:relevantEclassCategory_v5.1
    "AAB29200202"^^<http://www.w3.org/2001/XMLSchema#string>)
  value(ex:supportedBusinessFunctionsUNSPSC-BFI
    "14"^^<http://www.w3.org/2001/XMLSchema#int>))
)
```

```
Individual(ex:myCamera
  type(ex:ProductOrService)
  value(ex:relevantEclassCategory_v5.1
    "AAB29200202"^^<http://www.w3.org/2001/XMLSchema#string>
  )
)
```

We can also specify the product semantics by using eCl@ss **and** UNSPSC codes. However, we must keep in mind that this reflects the intersection of both product categories:

```
Individual(ex:CameraInvoice
  type(ex:ProductOrServiceRelatedEntity)
  value(ex:relevantEclassCategory_v5.1
    "AAB29200202"^^<http://www.w3.org/2001/XMLSchema#string>)
  value(ex:relevantUnspscCategory_v8
    "45121504"^^<http://www.w3.org/2001/XMLSchema#string>)
  )
)
```

All those annotations can be queried in a meaningful manner even without reasoning support, e.g. by SPARQL queries against RDF data. At the same time, they can be integrated into eClassOWL or similar ontologies, as shown in the next section.

4. Upward Compatibility: Mapping to eClassOWL

In this section, we show how annotations based on the lightweight approach proposed in this paper can be mapped to eClassOWL or similar products and services ontologies. This allows taking into account the subsumption hierarchy of eClassOWL and other features when interpreting such data. It is important to note that this mapping does not need to be created individually for each annotation. Rather, one may create one big mapping document for each relevant release of eCI@ss. This document can then be imported as needed. Since the mapping within the same classification standard (e.g. between eCI@ss literal values and eClassOWL classes) follows a simple schema, such document can be created fully automatically. The next release of eClassOWL will include such a mapping document as a separate OWL file. The same holds for the mapping between UNSPSC categories and UNSPSC ontology variants. More difficult is a mapping that bridges two or more such standards, e.g. from UNSPSC to eCI@ss [see e.g. 19]. This challenge is a research topic in its own right and not considered further in this paper.

It is important to know that eClassOWL has two types of classes for each category in the original classification: Firstly, one *generic* class (“gen”) that contains only *actual products* of that kind (i.e. an instance of “TV set” is a real product of that type), and secondly, a *taxonomic* class that contains “anything than can be subsumed under this label in any relevant context”. Those taxonomic classes also cover *related* products. These characteristics of eClassOWL and the underlying rationales are explained in more detail in [4] and [5]. For the following example, it is sufficient to know the general semantics of the generic and the taxonomic classes.

The class identifiers in eClassOWL are derived from the category identifiers by adding a preceding “C_” for “class” and adding a trailing “-gen” (C_XXXXXXXXXX-gen) for generic classes and “-tax” (C_XXXXXXXXXX-tax) for the taxonomic classes. Thus, the two classes for the eCI@ss category “Photo Cameras” with the original identifier “AAB29200202” have the eClassOWL identifiers “C_AAB29200202-gen” (for actual cameras) and “C_AAB29200202-tax” (for anything that is camera-related in any relevant context).

The mapping itself can be implemented in OWL by value restrictions on the respective datatype properties.

Accordingly, the mapping definition for the *generic* classes will look as follows in OWL Abstract Syntax:

```

Class (eClass51:C_AAB29200202-gen complete
  intersectionOf(
    ex:ProductOrService
    restriction(ex:relevantEclassCategory_v5.1
value ( "AAB29200202"@EN)))

```

The mapping definition for the *taxonomic* classes will differ in that it is based on the intersection of `ProductOrServiceOrRelatedInstance`, i.e. the more general concept from our lightweight ontology:

```

Class (eClass51:C_AAB29200202-tax complete
  intersectionOf(
    ex:ProductOrServiceOrRelatedInstance
    restriction(ex:relevantEclassCategory_v5.1
value ( "AAB29200202"@EN)))

```

Since value restrictions on datatype properties are outside of OWL DLP and OWL Lite, importing these mapping statements makes the resulting ontology reside in OWL DL. This is a bit of a disadvantage, since `eClassOWL` itself does not go beyond OWL DLP.

5. Evaluation

In the following, we compare the direct use of fully-fledged products and services ontologies with our lightweight approach and take into account practical and legal aspects. Our analysis considers the following dimensions:

1. **Size of the ontology:** How big is the ontology that needs to be imported when processing Web resources and their annotations?
2. **Upward Compatibility to eClassOWL:** Can the representation be mapped easily to fully-fledged products and services ontologies, namely `eClassOWL`?
3. **Ability to handle new types of products and services:** If additional products and services categories are needed, e.g. due to product innovation, how quickly and easily can they be agreed upon and used?
4. **Intellectual Property Issues:** Does the implementation of the approach require a legal agreement with the owners of existing e-commerce standards?

5. **Expressivity:** How much explicit and implicit knowledge about the products and services offerings can be gained from a respective annotation?

Table 3 gives a comparison of the two approaches.. One can see that our proposal of using category identifiers as datatype properties by means of the small ontology presented in section 4 has several appealing characteristics:

1. It does not require replicating the whole standard in the form of an ontology specification. This also eliminates the overhead for importing a large ontology when interpreting respective annotations.
2. It is easily upwards compatible with eClassOWL. In fact, the next release of eClassOWL will include respective mapping files.
3. It does not depend on an agreement with the standards owners prior to being allowed to publish the resulting ontology.
4. It can be used in a meaningful way even without the availability of a scalable reasoner, i.e. on the level of plain RDF data.

Table 3. Comparison of Approaches for the Representation of Products and Services

Approach Criterion	Ontologies derived from classification standards	Lightweight product description based on datatype properties
Size of the ontology	Huge (tens of megabytes)	Very small
Upward compatibility to eClassOWL	Simple (difficult may only be determining the correct relationship between product categories from <i>multiple</i> standards)	Simple (by value restrictions on the datatype properties)
Ability to handle new types of products and services	Limited (if new categories are needed, one must submit change requests to the respective standardization body; only after the standard has been updated, a new version of the ontology can be derived)	Limited (if new categories are needed, one must submit change requests to the respective standardization body; however, once the standard is updated, the new category identifiers are immediately usable – no delay caused by an ontology update). Note: Major release changes in the standards require new datatype properties.
Intellectual Property Issues	Problematic (explicit agreements with IPR owners required)	Simple (no agreements with IPR owners required as long as the usage of the codes themselves is granted to the general public, which is mostly the case)
Expressivity	Medium to High (depending on the amount of effort invested in lifting the standard to an ontological level)	Limited , but grounding in rich ontologies is straightforward

One might argue that the pure reference to flat concepts in external standards provides little help for reasoning about products in the Semantic Web. However, already a lot of gain in precision can be achieved by being able to search Web offerings by single UNSPSC or eCl@ss category codes. Also, our approach eases the development of intelligent applications significantly, because it decouples the two spheres of (1) product data annotation and (2) the development of richly axiomatized vertical ontologies for small product domains. In other words, we can annotate offerings already now with respective category codes, while the development of richer ontology variants of eCl@ss or UNSPSC may be an ongoing research challenge. Also, we assume useful lightweight applications can be developed right away.

A possible extension of our approach is to combine it with just the large property library provided by eClassOWL. This part of eClassOWL defines more than 5,500 datatype and object properties for typical product characteristics, like “weight”, “diameter”, “screen size”, etc. However, the property library of eClassOWL can currently not be separated from the class definitions for legal reasons.

One could also consider importing the vast amount of standardized properties from the DIN Properties Dictionary [33] in order to complement the lightweight approach with more specific product properties.

6. Conclusion

We have presented a straightforward solution for annotating offerings on the Web which avoids the overhead of importing fully-fledged products and services ontologies in any single annotation of a Web resource. The latter is unfeasible, since the respective ontologies are very large (20 – 70,000 classes), which makes them too large for frequent retrieval on-the-fly. Our approach does not require explicit legal agreements with the owners of popular classification schemas, while it can be easily mapped to fully-fledged ontology variants. The full ontology plus documentation will be released shortly at <http://www.heppnetz.de/prodlight/>. Also, the next release of eClassOWL will include mappings, as described in section 4, as an optional module, to be imported on demand. This will support interoperability between the ProdLight approach, meant for annotations embedded in Web resources and simple search, and heavyweight ontology applications that import eClassOWL and other large ontologies.

Acknowledgements: (Not included in the submitted version)

References

1. United Nations Development Programme: United Nations Standard Products and Services Code (UNSPSC). Available at <http://www.unspsc.org/> (retrieved May 22, 2006).
2. eClass e.V.: eCl@ss: Standardized Material and Service Classification. Available at <http://www.eClass-online.com/> (retrieved December 20, 2006).
3. Hepp, M.: The True Complexity of Product Representation in the Semantic Web. Proceedings of the 14th European Conference on Information System (ECIS 2006), Gothenburg, Sweden (2006): 1-12.
4. 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.
5. Hepp, M.: Representing the Hierarchy of Industrial Taxonomies in OWL: The gen/tax Approach. Proceedings of the ISWC Workshop Semantic Web Case Studies and Best Practices for eBusiness (SWCASE05), Galway, Irland (2005).
6. Hepp, M.: eCl@ssOWL. The Products and Services Ontology. Available at <http://www.heppnetz.de/eClassowl/> (retrieved December 20, 2006).
7. Samuelson, P.: Copyrighting Standards. *Communications of the ACM* 49 (6): 27-31, 2006.
8. Hepp, M.: Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies. *IEEE Internet Computing* 11 (7): 96-102, 2007.
9. Giunchiglia, F., Marchese, M., Zaihrayeu, I.: Towards a Theory of Formal Classification. Proceedings of the AAAI-05 Workshop on Contexts and Ontologies: Theory, Practice and Applications (C&O-2005), Pittsburgh, Pennsylvania, USA (2005): 1-8.
10. Giunchiglia, F., Marchese, M., Zaihrayeu, I.: Encoding Classifications into Lightweight Ontologies. Proceedings of the 3rd European Semantic Web Conference (ESWC 2006). Springer, Budva, Montenegro (2006).
11. United Nations Development Programme: Business Function Identifiers (BFI). Available at <http://www.un-spdc.org/AdminFolder/documents/BFI.doc> (retrieved December 20, 2006).
12. 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* (forthcoming) 2006.
13. Smith, H., Poulter, K.: Share the Ontology in XML-based Trading Architectures. *Communications of the ACM* 42 (3): 110-111, 1999.
14. Fensel, D.: *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, Berlin etc. (2004).
15. 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.
16. Zhao, Y., Sandahl, K.: Potential Advantages of Semantic Web for Internet Commerce. Proceedings of the International Conference on Enterprise Information Systems (ICEIS), Vol. 4, Angers, France (2003): 151-158.
17. Gupta, T., Qasem, A.: Reduction of price dispersion through Semantic E-commerce: A Position Paper. Proceedings of the Semantic Web Workshop 2002, Hawaii, USA (2002): 1-2.
18. Fairchild, A.M., de Vuyst, B.: Coding Standards Benefiting Product and Service Information in E-Commerce. Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS-35) (2002): 258b.
19. Schulten, E., Akkermans, H., Botquin, G., Dörr, M., Guarino, N., Lopes, N., Sadeh, N.: The E-Commerce Product Classification Challenge. *IEEE Intelligent Systems* 16 (4): 86-89, 2001.
20. 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): 54-59, 2001.

21. Omelayenko, B.: Ontology Integration Tasks in Business-to-Business E-commerce. In: Monostori, L., Vancza, J., Ali, M. (eds.): Proceedings of the Fourteenth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems. Springer, Budapest, Hungary (2001): 119-124.
22. Omelayenko, B.: Integration of Product Ontologies for B2B Marketplaces: A Preview. ACM SIGecom Exchanges archive 2 (1): 19-25, 2000.
23. Corcho, O., Gómez-Pérez, A.: Solving Integration Problems of E-commerce Standards and Initiatives through Ontological Mappings. Proceedings of the Workshop on E-Business and Intelligent Web at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, USA (2001): 1-10.
24. Yan, G., Ng, W.K., Lim, E.-P.: Product Schema Integration for Electronic Commerce - A Synonym Comparison Approach. IEEE Transactions on Knowledge and Data Engineering 4 (3): 583-598, 2002.
25. Obrst, L., Wray, R.E., Liu, H.: Ontological Engineering for B2B E-Commerce. Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS'01). ACM Press, Ogunquit, Maine, USA (2001): 117-126.
26. Omelayenko, B.: Preliminary Ontology Modelling for B2B Content Integration. In: Tjoa, A.M., Wagner, R.R. (eds.): Proceedings of the 12th International Workshop on Database and Expert Systems Applications. IEEE Computer Society, München (2001): 7-13.
27. Lee, H., Shim, J., Lee, S., Lee, S.-g.: Modeling Considerations for Product Ontology. In: Roddick, J.F., Benjamins, R.V., Cherfi, S.S.-S., Chiang, R., Claramunt, C., Elmasri, R., Grandi, F., Han, H., Hepp, M., Lytras, M., Misic, V.B., Poels, G., Song, I.-Y., Trujillo, J., Vangenot, C. (eds.): Proceedings of the First International Workshop on Ontologizing Industrial Standards, Vol. LNCS 4231. Springer, Tucson, Arizona, USA (2006): 291-300.
28. Zhao, Y., Lövdahl, J.: A Reuse-Based Method of Developing the Ontology for E-Procurement. Proceedings of the Nordic Conference on Web Services (NCWS), Växjö, Sweden (2003): 101-112.
29. Zhao, Y.: Develop the Ontology for Internet Commerce by Reusing Existing Standards. Proceedings of the International Workshop on Semantic Web Foundations and Application Technologies (SWFAT), Nara, Japan (2003): 51-57.
30. W3C: Simple Knowledge Organisation System (SKOS). Available at <http://www.w3.org/2004/02/skos/> (retrieved December 20, 2006).
31. Klein, M.: DAML+OIL and RDF Schema representation of UNSPSC. Available at <http://www.cs.vu.nl/~mcaklein/unspsc/> (retrieved December 20, 2006).
32. McGuinness, D.L.: UNSPSC Ontology in DAML+OIL. Available at <http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml> (retrieved December 20, 2006).
33. Deutsches Institut für Normung e. V.: DINsml.net - DIN-Merkmallexikon (DIN Properties Dictionary). Available at <http://www.dinsml.net/> (retrieved December 20, 2006).