In: Manolopoulos, Y.; Filipe, J.; Constantopoulos, P.; Cordeiro, J. (Eds.): Enterprise Information Systems VIII, Springer LNBIP Vol. 3, Berlin, 2008, pp. 363-374.

Controlling Property Growth in Product Classification Schemes: A Data Management Approach

Joerg Leukel[1]

[1] University of Hohenheim, Information Systems II, Schwerzstr. 35,
70599 Stuttgart, Germany
joerg.leukel@uni-hohenheim.de

Abstract. Product classification schemes aim at semantic interoperability in B2B e-commerce by providing consensual definitions of product categories and recommending properties for describing product instances. Considerable industry work has been carried out on enhancing the size and thus coverage of these schemes. Horizontal classification schemes, however, often consist of more than 10,000 classes, several thousand properties, and an even greater number of class-property relations. The problem is that maintaining these schemes becomes more and more demanding in particular due to the number of definitions and interrelations. This paper proposes measures for coping with the problem of extensive and steadily growing property libraries. We view these schemes from a data modeling perspective and relate the proposed measures to the underlying conceptual data model of product classification schemes. It can be shown that these measures greatly influence both standards makers and standards adopters.

Keywords: B2B, E-commerce, Product Data Management, Product Ontologies, Standardization.

1 Introduction

Executing business processes between independent organizations faces often heterogeneity concerning process models, data sources, software systems, and available meta data describing these components. Automating such processes increases the need for aligning heterogeneities and finding consensus about common concepts. Ontologies aim at fulfilling this role by establishing a shared and common understanding of a domain. In B2B e-commerce, most processes incorporate essential information about products (and services) being the subject of procurement and sales respectively. Therefore, the development of product ontologies can be regarded as an enabler of machine-readable, unambiguous representations of information about products [1] [2].

Many industry consortia have proposed such domain ontologies called standard product classification schemes (standard PCS). Applying these business vocabularies benefits searching for products in e-catalogs, comparing similar products, standardizing product descriptions, and facilitates spend analysis and productsensitive workflows [3]. Prominent horizontal standards, such as eCl@ss, eOTD and UNSPSC consist of 20,000 up to 60,000 product classes, and represent a huge amount of knowledge about the categorization of products.

For standards makers, the broad coverage of horizontal standards leads to an enormous amount of properties as new subjects of standardization including proposal, negotiation, definition, and maintenance. Taking in mind the distributed nature of many standardization processes (e.g., work groups for each sub-domain or branch of industry), reducing or avoiding redundant properties becomes an important task. For instance, work groups should always check carefully the appropriateness of existing properties before proposing a new property for the vocabulary. This basic principle does not affect the underlying model, but the standardization process. It requires, however, that properties are reusable. Reusability of properties depends on their semantic precision, naming issues (i.e., synonyms, homonyms), and the conflict between wide or narrow definitions.

For standards adopters, properties must be seen from a different perspective. Classification based on a standard PCS requires (1) assigning each product to a class and (2) describing each product with class-specific properties. This initial effort is time-consuming and costing; it depends mainly on the number of products and the number of properties per class. In addition, it requires continuous efforts triggered by new PCS versions (new product classes, new properties, redefinition of classes and properties).

This paper proposes measures for coping with the problem of extensive and steadily growing property libraries. We view these schemes from a data modeling perspective and relate the proposed measures to the underlying conceptual data model of product classification schemes. Our contribution to research is that we (1) introduce the still overseen problem of property growth and (2) provide a comprehensive set of measures that address this problem.

The remainder of our paper is structured as follows. Section 2 discusses related work, and shows that extensive property libraries have rarely been the subject of research. In section 3, we describe the main problem based on empirical observations. In addition, we present the basic conceptual data model of property-centric classification schemes, which will be extended in the course of our work. In section 4, we define measures for coping with property libraries, and describe their impact on the problem. In section 5, we discuss the measures by summarizing their impacts and extending the basic model. Finally, we draw conclusions and point to future avenues of research.

2 Related Work

Related work to product classification schemes can be found in several fields such as e-catalogs, product data management, standardization, and ontology engineering. Next, we provide an overview of closely related work and outline their relevance to the problems caused by extensive property libraries.

Early work on basic concepts of PCS presents and evaluates standard PCS from a business perspective. For instance, [4] examined the role of standard PCS towards benefits of spend analysis; properties are not necessary for this business function.

Schulten et al. introduced product classification as a reference domain for ontology engineering and the Semantic Web, and called for concentrated efforts to "design a generic model" for automated mapping between two different PCS [5]. Concerning product properties, the proposal demonstrated the mapping problem between classes only, but did not incorporate properties. Eventually, the research prototypes in [6] and [7] followed this class-centered path.

Ng et al. described challenges in integrating product schemes based on heterogeneous properties [8]. Property lists form schemas that can be integrated by applying techniques from database schema integration. Two interesting aspects discussed by Ng et al. are shallowness (flat structures, lists) and bushiness (clusters of a high number of related properties). This database approach is complemented and extended in [9], which analyzed property mappings and practical issues in more detail.

Leukel et al. proposed an XML-based exchange format for PCS [10]. Its contribution lies in identifying and defining data elements and relationships, both being derived from an empirical study of four standard PCS and three XML e-business standards. The modeling of properties is quite sophisticated and fulfills mainly requirements of PCS adopters. Two measures for supporting "management and maintenance" of property libraries – grouping and inheritance – are described briefly.

Recently, the importance of properties as a cornerstone of product classification has become more evident. For instance, Ondracek and Sander [11] argued on a "property based product classification" from that multiple different classification hierarchies for specific purposes can be built, though they are based on common, thus standardized properties. Leukel [2] emphasized the role of properties for providing additional semantics to class hierarchies; properties are needed to describe the scope of a class formally. Kim et al. developed a "semantic classification model" [12] based on properties in order to enable an in-depth understanding of product classification. All this work is in support of semantically rich PCS that incorporate well-defined properties. The problems caused by large property libraries are being overseen though.

A first indication of problems related to properties can be found in early work by Hepp [13]. The proposed quantitative measurements for PCS reveal some shortcomings in property lists and can help detect duplicate properties. In its conclusion, the paper argues on the need for further work on maintaining properties and organizing property libraries. A comprehensive quantitative analysis of classes and properties in PCS is subject of [14]. In addition, Ondracek and Sander drew attention to the problem of redundant properties in huge property libraries, and claimed that separating definition and application of properties would be "the only solution" [11].

3 Problem Definition: Property Growth

In recent years, considerable industry efforts have been undertaken to extend the semantic richness of product classes by adding class-specific property lists. A property list contains all properties that should be used to describe products belonging to the respective class. These lists greatly enhance the formal precision of standard PCS by replacing class labels with a structured, though human-language description of its meaning. From an ontology perspective, property lists can be regarded as a first step towards true product ontologies, since they provide standardized representations of product concepts, thus machine-readable semantics [14]. Defining property lists not only requires resources, but it also causes new problems for both standards makers and standards adopters. We refer to these problems as of *property growth*.

3.1 Observations

Property growth can be assessed by comparing multiple versions of the same standard PCS. Next, we present some data for eCl@ss, a horizontal PCS being developed by a consortium of mainly German companies since the late 1990s [15]. It has gained a significant relevance for e-procurement in many European countries. Table 1 shows basic figures for five versions of eCl@ss.

Table 1. Growth of Classes and Properties in eCl@ss.

	V4.0	**V4.1**	**V5.0**	**V5.1**	**V5.1.1**
Publication Date	Aug 2001	Sep 2002	Dec 2003	Sep 2004	Sep 2005
4th Level Classes	10,190	12,565	20,379	21,100	22,203
Properties	2,303	5,504	3,667	5,525	6,941
Class-Property Relations	68,244	303,511	406,482	403,859	440,430

The number of classes, properties, and class-property relations has tremendously increased (118%, 201% and 545%) in the past four years. The increase of properties, however, is not constantly over time (significant decline in V5.0). We can assume that eCl@ss had implemented some actions to limit redundancy by reorganizing the property library, although, the sheer number of property remains high.

Looking closer at eCl@ss (table 2), we observe that property lists were added to more and more classes and most property lists contain at least 30 properties (V5.0: 83% of all property lists; V4.1: maximum of 294 properties!).

The growing number of property lists and properties per class causes significant classification costs, thus calls for property-centric classification strategies for PCS adopters (i.e., suppliers, buyers, marketplaces). Moreover, we have to consider that in many industries multiple standard PCS are available, especially due to competition between horizontal standards. Each standard PCS defines its own classes, properties, and property lists. Therefore, the problem of property growth is multiplied by the number of standard PCS.

Table 2. Property Lists in eCl@ss.

	V4.0	V4.1	V5.0	V5.1
Classes with Property Lists	1,107 (10.9%)	6,507 (51.8%)	7,913 (38.8%)	10,930 (51.8%)
Properties per Class: Minimum	6	6	3	1
Properties per Class: Maximum	89	294	266	156
Properties per Class: Mean	20.6	42.0	43.5	32.3
Properties per Class: Derivation	10.3	16.7	13.3	15.2

3.2 Basic Model

Product classes are the core components of each PCS. A product class is a categorization, collection or type of similar products that share a set of characteristics (e.g., the class 'laptop' describes portable computers). Product characteristics are expressed by properties (e.g., CPU type, display size, weight). Properties are not limited to a single class, but should be reusable. While some properties represent information that can be captured by standard data types (e.g., string, integer, float, Boolean), other properties limit the allowed values to a specific list of values (e.g., color 'red', 'green', 'blue' etc.). In conceptual modeling, the definition of enumerated domains can be expressed by a tertiary relation between product class, property and value as it is shown in figure 1. This data model introduces elementary attributes for each entity type. For instance, a property consists of its identification, name, textual definition, data type, and unit of measurement (UOM, e.g., meters, kilogram, and volt). Similar models are used in [9], [10], and [11].

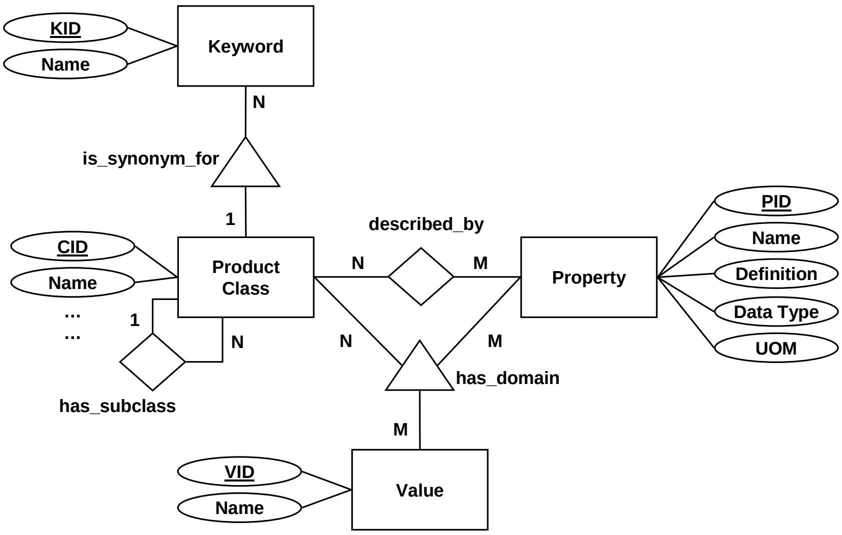

Figure 1. Basic Conceptual Data Model.

The major drawback of this model can be illustrated by a simple example: In industry segment A, which is represented by a number of classes, the property ‘length’ is measured in inches, while industry segment B – being represented by other product classes of the same sub-tree – measures the very same property in meters (e.g., hand tools vs. pipes for gas transportation). Consequently, two properties must be defined; their specification is nearly equal and differs only in UOM. This is especially for horizontal PCS, which cover a broad range of industry segments, a common problem. It can be solved by increasing the reusability of properties, though this may require modifications of the model.

4 Data Management Measures

In this section, we define a comprehensive set of measures that can be taken by standards makers. These measures address data management issues and relate to the underlying conceptual data model. We describe the rationale and, if necessary, point out to extensions of the basic model.

For each measure we assess the potential on the given problem. There are ‘two sides to every story’ – standards must be developed and maintained, and standards should be applied; otherwise they would be no standards. The effect on standards makers concerns changes in initial efforts (i.e., setting up the PCS) and maintenance efforts (i.e., processing change requests, releasing new versions). Similarly, the effect on standards adopters refers to the initial classification process and subsequent re-classification processes. Another aspect of PCS adoption is the GUI presentation and actual usage in applications such as e-procurement, marketplaces, and product data management systems. This aspect will be considered, too.

4.1 Maximum Number of Properties

Rationale: The number of properties per class is limited to a fix number (e.g., 15); this limitation applies to all property lists, thus to the entire PCS. It prevents property growth locally, especially with regard to product segments in which product descriptions can be very detailed. This measure does not modify the basic model, but adds a constraint on the cardinality of the class-property relation.

Standards makers: The implementation of this measure requires making a single decision on the maximum number. Eventually, the size of the property library is limited as well (only by the number of classes and the reuse of existing properties). There is even a significant change in maintenance efforts, since adding a new property to a property list is not possible if the maximum number has already been reached.

Standards adopters: Both the classification and re-classification efforts are reduced and can be forecasted. The GUI representation is improved due to lower space requirements and may fit on a single screen in all cases (e.g., imagine the list of 294 properties compared to the reduced list of the 15 most important properties).

4.2 Optional Properties

Rationale: Properties are distinguished whether their usage is optional or mandatory. This distinction aims at reducing the number of essential properties that have to be used for product description, while it does not remove properties from property lists. The number of optional and mandatory properties depends on the product class; for instance, all properties may be optional, or mandatory (min/max-approach). This measure adds a new attribute 'mandatory' to the class-property relation.

Standards makers: For each property of each property list, the question of mandatory or optional has to be answered. This initial effort can be reduced by setting all properties to optional followed by searching for the most important, thus mandatory properties. Because of product innovation, optional properties may be shifted to mandatory and vice-versa (maintenance effort).

Standards adopters: Depending on the share of optional properties, the classification effort is reduced. This measure allows diverse classification strategies, i.e., support only mandatory properties. Adopters decide on supporting optional properties, especially if these are required by their customers. Moreover, there are two consequences on GUI representation: (1) optional properties can be marked, thus separated from mandatory properties, and (2) parametric, property-based search for products has to be restricted to mandatory properties.

4.3 Naming Conventions

Rationale: The name of a property must adhere to specific naming conventions in order to prevent redundant properties (e.g., 'diameter, max.' vs. 'maximum diameter' vs. 'max. diam.'). This measure addresses the problem of finding the right property in the property library, in addition to keywords. Types of conventions: prefix vs. postfix qualifiers, singular vs. plural, use of abbreviations, separation of UOM from property name (e.g., 'diameter' instead of 'diameter in mm'). This measure does not modify the basic model.

Standards makers: Initially, naming conventions must be developed, and applied to all property names. Additional property names to the harmonized name can be stored in the keyword list. Applying these conventions can reveal redundant properties that should be removed from the property library.

Standards adopters: The classification process is not directly affected; searching for the right property is slightly improved. Since the naming can be used to build logical groups of properties, the GUI representation of large property lists is improved. For instance, postfixes to property names (e.g., 'length, max.', 'length, min.') express a specialization of closely related properties.

4.4 Property Groups

Rationale: Each property belongs to a predefined group (e.g., design, dimensions, shape, and business properties). This categorization eases the handling of huge property lists, since the flat list is transformed into a hierarchical structure. The basic model has to be modified: define a list of groups, and add a N:1 relation between property group to property.

Standards makers: Implementing this measure requires defining non-overlapping groups and assigning each property of the property library to one group. The maintenance effort is slightly effected (assign each new property to one group). The subdivision of the property library helps overlooking all properties, though it does not affect the total number of properties.

Standards adopters: Similarly to naming conventions, this measure does not influence the classification process. In the same way, it improves the GUI representation by explicitly defined groups of similar properties.

4.5 Views on Property Lists

Rationale: Instead of defining a single comprehensive property list, define overlapping views on property lists for each stage of the product lifecycle. The rationale is that the relevance of a property depends mainly on the product lifecycle and the respective business function of product data. For instance, the requirements of spend analysis differ from those of engineering. In consequence, each view-specific property list can be reduced to purely relevant properties. Eventually, the class-property relation must be modified to reflect the view.

Standards makers: Views on property lists result in multiple, overlapping property lists for the same product class; hence the initial and maintenance efforts are considerably higher. While the number of properties in those lists is reduced because of including only view-relevant properties, the total number of properties in the property library remains unchanged.

Standards adopters: Due to strictly view-relevant properties, the efforts for classification can be reduced in those cases where not all product lifecycle phases are relevant. Often, standard PCS are only used for procurement or sales and not for intra-organizational purposes; therefore, this benefit is quite relevant. In addition, the GUI representation no longer contains non-relevant properties.

4.6 Property Templates

Rationale: Instead of defining properties completely, the property library contains templates only. These generic properties can be used for multiple specific purposes by concretizing the template. This concept is very similar to separating property definition and property application as described in [9]. For instance, the template includes name and definition, while the concretization adds data type and UOM. Regarding the basic model, the class-property relation is extended by further attributes that were formerly part of the property entity type.

Standards makers: The first step for implementing this measure is deciding which attributes still belong to the generic property and which attributes belong to the class-specific property, thus to the class-property relation. Eventually, a rather small number of generic properties needs to be defined from which more specific properties can be instantiated. Concerning maintenance, adding a new property can often be based on a similar, already existing property (i.e., concretizing the generic property instead of defining the new property completely).

Standards adopters: This measure does not influence the classification process nor does it improve the GUI representation. The reason is that property templates concern only the organization of the property library.

4.7 Property Inheritance

Rationale: So far, all measures were directed at properties only. Considering that properties are assigned to product classes forming a class hierarchy, property inheritance says that properties are inherited to all lower classes. Moreover, an inherited property can be modified (concretized) on lower levels. This measure does affect the basic model as follows: the class-property relation as well as the class-property-value relation is available for all classes, not only for leaves of the class tree.

Standards makers: Setting up a PCS based on property inheritance calls for thoroughly defined properties that can be assigned to nodes of the class tree; otherwise the benefits of inheritance will not be realized. Moreover, the class hierarchy itself has to be suitable for assigning properties that are common for complete sub-trees. Maintaining such a PCS requires fewer efforts, since the property library contains lesser properties and sub-trees truly represent similar product classes characterized by a set of common properties.

Standards adopters: Similarly to property templates, this measure concerns the property library only.

5 Discussion

In this section, we discuss the proposed measures by summarizing the expected effects and modify the basic model.

5.1 Summarization of Effects

Next, we compile the previously assessed effects of each measure on the problem of property growth (table 3). For both standards makers and adopters, we state expected changes regarding initial and maintenance efforts (‘-‘ for decrease; ‘+’ for increase, ‘o’ for no change). Effects on the total number of properties and GUI representation are further criteria of our assessment (‘reduced’ and ‘improved’ respectively).

Comparing the effects on standards makers and adopters, we have to state that 6 out of 7 measures increase the initial effort for standards makers, while classification efforts are reduced or remain unchanged; a reduction of the number of properties can be expected for four measures, while the GUI representation is improved by five measures.

Table 3. Effects on Standards Makers and Standards Adopters.

	Standards Makers			Standards Adopters		
Measure	**Initial Effort**	**Mainten-ance**	**Number of Properties**	**Initial Effort**	**Mainten-ance**	**GUI**
Maximum Number	-	-	reduced	-	-	improved
Optional Properties	+	o	o	-	o	improved
Naming Conventions	+	o	reduced	o	o	improved
Property Groups	+	o	o	o	o	improved
Views on Prop. Lists	+	+	o	-	-	improved
Property Templates	+	-	reduced	o	o	o
Property Inheritance	+	-	reduced	o	o	o

5.2 Modification of the Basic Model

Since the proposed measures concern the definition of single properties being elements of a property library, the basic model for PCS needs to be modified. Therefore, we collect the modifications described before and alter the basic model as shown in Figure 2. The modifications include (1) adding one new attribute ('Mandatory'), (2) adding two new entity types (property group, view), (3) redefining one relationship ('described_by'), and (4) moving attributes from the property entity type to the described_by relationship (here: data type, uom).

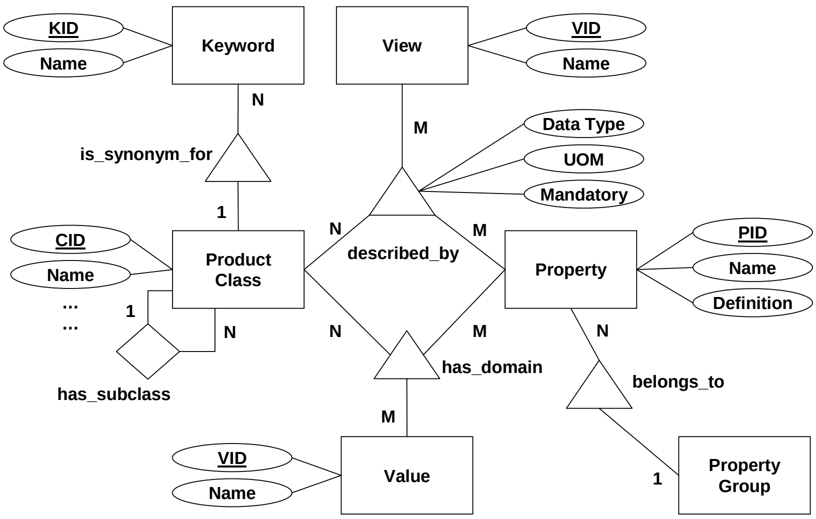

Figure 2. Extended Conceptual Data Model.

6 Conclusions

The main contribution of this paper lies in proposing new measures for coping with the problem of extensive and steadily growing property libraries. The assessment of their impact on standards makers and standards adopters revealed that the effects concern not only the total number of properties, but also initial efforts, maintenance efforts, and GUI representation issues. Therefore, decisions on implementing these measures should bear in mind all these criteria. The results of the assessment as well as the modified model may serve standards makers in their decision process on reorganizing property libraries.

The quantitative analysis (see section 3.1 and also [14]) of the property library in eCl@ss has drawn attention to the problem of defining, maintaining and actually using huge sets of product properties. While this quite elaborated standard PCS claims to be unique in its property-centered approach, the conceptual model of its property library is rather simple. We conclude that eCl@ss still focuses on semantic richness (i.e., extending the coverage of industry segments) rather than formal precision and efficient maintenance. This example, nevertheless, underlines the need for re-thinking the current organization of property libraries, since measures tackling the described problems are at hand, their impact can be predicted, and some of the proposed measures have already been tested in vertical standards.

Considering recent developments in finding consensus about the basic components, underlying conceptual models as well as maintenance policies of PCS, standards bodies and industry consortia have joined efforts in harmonizing their proprietary approaches in several initiatives and on different levels of obligation. For instance, the CEN project on product classification [3] states that a "good" PCS necessarily incorporates properties and property lists. Standard PCS that are purely based on classes are expected to add properties in order to broaden the range of application and to provide extended semantics.

Concerning the current state of standard PCS, these transformation processes will be quite demanding. From this point of view, we plan future work on validating the measures by quantitative analysis of standard PCS, thus reengineering their property libraries based on automated conversion, and developing transformation strategies for standards makers. Another field of interest derives from the role and suitability of reference models for PCS. These models will become more important since many standard PCS aim at converting their proprietary data models to the ISO 13584 standard [16]. This standard requires some modifications to the property library, though it does not address the maintenance problem explicitly as described in this paper, since its main purpose is to achieve semantic interoperability between different property libraries. Therefore, we see the need for extending the scope of this reference model to content management issues that greatly determine the costs and efforts of defining and implementing respective standard PCS.

References

1. Fensel, D. et al.: Product Data Integration in B2B E-Commerce. IEEE Intelligent Systems 16, 54--59 (2001)
2. Leukel, J.: Standardization of Product Ontologies in B2B Relationships – On the Role of ISO 13584. In: 10th Americas Conference on Information Systems, pp. 4084--4091. AIS (2004)

3. CEN/ISSS: CEN Workshop Agreement 15925:2005 – Description of References and Data Models for Classification. Brussels (2005)
4. Fairchild, A. M., de Vuyst, B.: Coding Standards Benefiting Product and Service Information in Ecommerce. In: 35th Annual Hawaii International Conference on System Sciences, pp. 258b. IEEE Computer Society, Los Alamitos (2001)
5. Schulten, E. et al.: The E-Commerce Product Classification Challenge. IEEE Intelligent Systems 16, 86--89 (2001)
6. Corcho, O., Gómez-Pérez, A.: Solving Integration Problems of E-commerce Standards and Initiatives through Ontological Mappings. In: Workshop on E-Business and Intelligent Web at the 17th International Joint Conference on Artificial Intelligence (2001)
7. Beneventano, D. et al.: A web service based framework for the semantic mapping amongst product classification schemas. J. of Electronic Commerce Research 5, 114--127 (2004)
8. Ng, W., Yan, G., Lim, E.-P.: Heterogeneous product description in electronic commerce. ACM SIGeCom Exchanges 1, 7--13 (2000)
9. Bullig, A., Schnadhorst, T., Wilkes, W.: Mapping of product dictionaries and corresponding catalog data. In: 10th ISPE International Conference on Concurrent Engineering, pp. 225--234. Balkema Publishers (2003)
10. Leukel, J., Schmitz, V., Dorloff, F.-D.: A Modeling Approach for Product Classification Systems. In: 13th International Workshop on Database and Expert Systems Applications, pp. 868--874. IEEE Computer Society, Los Alamitos (2002)
11. Ondracek, N., Sander, S.: Concepts and benefits of the german ISO 13584-compliant online dictionary www.DINsml.net. In: 10th ISPE International Conference on Concurrent Engineering, pp. 255--262. Balkema (2003)
12. Kim, D. et al.: A semantic classification model for e-catalogs. In: 6th IEEE International Conference on E-Commerce Technology, pp. 85--92. IEEE Computer Society, Los Alamitos (2004)
13. Hepp, M.: Measuring the Quality of Descriptive Languages for Products and Services. In: Multikonferenz Wirtschaftsinformatik, pp. 157--168. Cuvillier, Goettingen (2004)
14. 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 (2007, forthcoming)
15. eCl@ss e.V.: eCl@ss – Standardized Material and Service Classification, http://www.eclass-online.com
16. ISO: ISO 13584-1:2001 Industrial automation systems integration -- Parts library -- Part 1: Overview and fundamental principles. Geneva (2001)