

AUTOMATING PRODUCT CLASSIFICATION: THE IMPACT OF STANDARDIZATION

Joerg Leukel

*Institute for Computer Science and Business Information Systems (ICB), University of Duisburg-Essen
Universitaetsstr. 9, 45117 Essen, Germany
joerg.leukel@uni-essen.de*

ABSTRACT

Approaches to automating product classification adopt mainly algorithms and tools for classification in general, such as Vector Space Model, Bayesian Classification and K-Nearest Neighbor Classification, or take classification systems as (database) schemas that have to be integrated. Much work has been carried out on applying, evaluating and improving these tools for this specific domain, product data in B2B e-commerce. The formal specification of a product classification system (PCS) is seen as input data; hence this data has to be imported and often converted into an internal representation. This paper takes a closer look at this input data and examines its structure, semantic richness and degree of standardization. The reason is that standards for product classification systems and standardized specifications address the data exchange issues of these systems – and more important, are able to increase the success of automated product classification. To do so the paper will derive and analyze major standardization trends, show their interdependencies and evaluate their impact on automated classification.

KEYWORDS

B2B, E-Commerce, E-Procurement, ISO 13584, Standardization, XML

1. INTRODUCTION

Product classification in B2B e-commerce has the task to assign each product of an electronic product catalog to a class of a given product classification system (PCS). All products within a class fulfill a similar function and/or have similar attributes, thus they are similar or equivalent to each other. Classified product data is a success factor in web-based procurement such as desktop purchasing systems and electronic marketplaces. This is especially true for *standard product classification systems* (e.g., UNSPSC, eCI@ss, eOTD, NCS, RNTD) which enable efficient catalog navigation and product search as well as qualified product comparison across suppliers, since their class definitions are supplier-independent (Schulten et al., 2001; Fensel et al., 2001).

Classifying product data is a time-consuming and error-prone effort which requires extensive knowledge of the product domain. This process can be partly automated by adopting algorithms for classification in general. The Vector Space Model (VSM) is such a classic method of information retrieval (IR). Other methods are Bayesian and K-Nearest Classification. These IR methods have in common that they determine similarities between a search query (here: product data) and a set of documents (here: classes). A fundamentally different approach is schema integration which takes two product classification systems S1 and S2 as two schemas which have to be integrated by determining mappings between classes. If a product belongs to class N of system S1 and this class is mapped to class M of system S2, then the product belongs to class M, too; thus the classification process can be automated.

While automated product classification is quite good described in theory, there are significant problems caused by basic conditions of the product data domain and obstacles in handling product classification systems. These include different concepts for building classification systems, a low semantic level of information describing classes and their relationships, and different vocabularies for supplier-drenched product data vs. supplier-independent class hierarchies. The last two aspects reduce the success of automated

product classification, since IR as well as schema integration are mainly based on terminological similarities rather than formalized semantics.

2. PAPER ORGANIZATION AND RELATED WORK

Instead of working on domain-situated classification algorithms, this paper takes a closer look at the input data for automated classification, thus the PCS itself, and examines its structure, semantic richness and degree of standardization. The reason is that standards for PCS improve the data exchange issues, and more important, are able to increase the success of automated product classification. Our work may help to improve classification approaches by assessing the current state and emphasizing future directions of relevant standards. In addition, through our empirical analysis we describe needs for extended standards and recommend a stronger acceptance of common standards by classification bodies, thus the organizations that develop and maintain product classification systems. To do so, the remainder of our paper is structured as follows: in Section 3 we will discuss approaches to automating product classification. In Section 4 we will derive and analyze major standardization trends from industry practice. By relating standardization trends and classification approaches we will determine the impact of standardization on automated classification in Section 5.

Much work has been carried out in applying, evaluating and improving classification tools for this specific domain, product data in B2B e-commerce. The Vector Space Model for product classification is implemented in GoldenBullet (Ding et al., 2002). A wrapper imports all PCS specification data; it is limited to UNSPSC and class definitions. Contrary to the extended specification the classification process is based on class names only. The problem of different data models and extended specification is not described.

An implementation of the schema integration approach is the MOMIS system (Bergamaschi et al., 2002). MOMIS allows defining relationships between classes by (1) comparing class names and relating them to terms of the WordNet ontology, (2) building relationships manually by domain experts, and (3) deriving additional relationships through inference algorithms. The problem of schema heterogeneity as described in our paper is handled by a flexible wrapper, thus MOMIS does not concentrate on a common structural model including attribute definition (no extended specification, no mappings between attributes).

Similar to our approach, (Leukel et al., 2002) describes structural aspects of PCS. However, its aim is to develop an universal, XML-based exchange format. It derives requirements from an analysis of four PCS (eCI@ss, EGAS, RNTD, and UNSPSC) and three exchange formats (BMEcat, OAGIS, and xCBL). There are two major limitations: ISO 13584 is not mentioned, and the role of standardization that supports and facilitates automated product classification is neglected.

3. BASIC APPROACHES TO AUTOMATING PRODUCT CLASSIFICATION

The most simple classification procedure compares product descriptions (catalog) and class definitions (classification system) directly. This can be done by comparing product names with class names for example. However, such a procedure will not result in any worth mentioning classification success. For instance, (Ding et al., 2002) have tested this with real-world catalog data and they report of 0,2% products which have been classified correctly to the UNSPSC classification system. Reasons are the highly different vocabularies and naming principles for products and classes. Another reason is the limited range of product descriptions and class definitions. It is not confined to a pair of just two data elements (here: product and class names). Additional elements like describing texts, product attributes and synonyms have to be considered, too.

3.1 Information Retrieval

The first group of product classification approaches adopts methods of information retrieval. These methods have in common that they address domain-independently the question, how to determine required information from a set of semi-structured information by a search query. The central concept of IR is

document-orientation. Classification can be seen as a search problem referring to a set of documents. The product that has to be classified constitutes the search query, while the classification system forms the set of documents; each class forms a document of this set. The search query covers all information that describes the product. A common IR method for classification is the Vector Space Model (VSM), which represents the search query and the documents as vectors. The vectors develop from previously determined attributes and their values. Therefore the similarity of search query and document can be expressed by the angle between the two vectors. By applying the cosine function, this angle is transformed into a value between 0 and 1, thus it is a good measure of similarity. Applying the VSM to product classification requires some configuration steps which highly determine the classification success.

Choosing the right attributes is the most important parameter for the classification success, because the similarity measure of VSM is based on these attributes. If we say attribute in this context we mean the occurrence of a term in a document. These attributes must not be confused with product attributes. Only those attributes should be used that make a high contribution to identifying similarities or differences. This contribution is not made by all terms of a set of documents.

Scaling attributes has the task to map the values of attributes for all documents to a unified numeric domain. Since the VSM only distinguishes the appearance and non-appearance of a term, it is evident to normalize the domain of values to an interval between 0 and 1, where the value 1 means that the respective term appears in the document, and the value 0 means, that the term does not appear. Further, it is common to assign staged values in dependence on the position of the term in the document, e.g. 0; 0.25; 0.5; 0.75 and 1.

Defining the data base of the VSM comprises those document parts that have to be searched for terms. Applied to product classification, the data base has to be formed as a subset of product data. It is a subset, because some product data contains no terms at all, but coded information (e.g. product number, order unit, price, price currency, and availability). Other product data does not make any contribution to the classification problem, because the included terms do not relate to products (e.g. manufacturer name, contact information).

3.2 Schema Integration

The second group of classification approaches requires that all products, which have to be classified, are already classified by another classification system; thus these products are pre-classified. This approach aims at avoiding direct mappings between products and classes; instead it develops relationships between classes of two product classification systems. If we know these relationships then it is just a simple and automated step to classify any product data which fulfills the requirement described above. Integrating two classification systems can be seen as a problem of schema integration, where the classification systems are two different (database) schemas. In the following we describe the integration requirements that arise from the product classification domain.

The integration approach is characteristic for a situation where (1) the product catalog uses a supplier-specific PCS and (2) the supplier has to support a given (standard) classification system in order to participate in a marketplace or to fulfill a requirement of a customer. What we need here is a re-classification of pre-classified product data.

The data base of product classification integration is formed by two classification systems and their relating data describing classes. Very similar to data transformations and schema mapping we have to define mappings, but they relate classes to each other, not data elements (as parts of schemas). In this terminology a mapping is a semantic relationship between one or more classes of a source classification system and one or more classes of a target classification system. The mapping types can be differentiated by looking at the number of classes participating in the source and target system, thus the cardinality of the mapping. This concept leads to six mapping types: 1:1, 1:N, N:1, 1:0 and 0:1 as described in table 1. Its third column states what effect such mappings have on automating the classification of product data. The N:M mapping reveals that the classification systems to be integrated are based on two different structuring approaches; thus the classes are built on complete different criteria. This mapping occurs when integrating an application-oriented and a feature-oriented PCS.

Table 1. Mapping Types for integrating Product Classification Systems.

Cardinality	Definition	Re-Classification Process
1:1	One class of the source system is equivalent to one class of the target system.	Automated
1:N	One class of the source system is equivalent to two or more classes of the target system.	Semi-automated; target class has to be selected manually
N:1	Two or more classes of the source system are equivalent to one class of the target system.	Automated
N:M	Two or more classes of the target system are equivalent to two or more classes of the target system. The mapping is not decomposed into 1:1, 1:N or N:1 mappings.	Semi-automated; target class has to be selected manually
1:0	The class of the source system has no equivalent in the target system.	Classification not possible at all
0:1	The class of the target system has no equivalent in the source system.	(not relevant)

4. STANDARDIZATION TRENDS IN PRODUCT CLASSIFICATION

In this Section we concentrate on standardization as one of two basic instruments to enhance interoperability, the other being integration. For terminology reasons we first must distinguish the following terms:

- A *standard PCS* is developed by organizations for a specific domain or across branches of industry. In this case, the term standard relates to the content of the PCS, meaning the hierarchy of classes and attached attributes.
- A *standardized specification* is a specification of a PCS that adheres to a specification standard. This standard may cover the conceptual level and/or the exchange format level.
- A *standard exchange format* is a format for exchanging PCS and is developed by standardization organizations (companies, consortia, associations or standardization bodies such as ANSI and ISO).

4.1 Exchange Formats

To be able to classify a product, companies need to know the PCS, thus the data describing the system must be imported into an information system, e.g., catalog management system. This role is fulfilled by the system definition, which is a set of structured data that describes and defines the content of the classification system. Importing system definitions requires that this data can be exchanged between participating companies. Companies that apply such a system require these system definitions.

PCS and their system definitions are subject of all organizations participating in catalog-based transactions; PCS are a part of product catalogs, or the exchange process is separated from catalogs. Especially in the case of standard PCS it is important that all market partners are able to access and process the system definitions. From the view of organizations that develop standardized systems it must be guaranteed that suppliers, intermediaries and purchasing companies are supplied with the definitions in a given exchange format. In this regard, one could think that the complexity of data exchange and the number of exchange formats are already reduced by using standard PCS. However, business practice does not prove this.

In general, each standard PCS uses a different exchange format. Consequently, importing these definitions calls for flexible tools for mapping the data elements of the exchange format to the importing information system. Heterogeneity is not limited to the general format type such as comma separated values (CSV), Microsoft Excel spreadsheets (XLS) and Microsoft Access database files (MDB) but addresses the number of data elements, their naming and relationships. Thus the schemas are different and have to be integrated or mapped to a common schema.

Standardization in this area is two-fold: on one hand, several XML-based standard exchange formats for product catalogs are available, and these formats also cover the exchange of PCS. Catalog formats such as BMEcat (Schmitz et al., 2002), cXML (Ariba, 2004), OAGIS (Open Applications Group, 2002) and xCBL (CommerceOne, 2002) have to be mentioned. However, the analysis in (Leukel et al., 2002) has shown that

their suitability for transferring standard PCS is very limited, because relevant information losses occur, especially regarding attribute definitions.

On the other hand, we have to consider the ISO 13584 standard which aims at serving as a reference model for PCS (ISO, 1998). ISO 13584 originates in the product data management area and is therefore focused on engineering, construction and production of technical goods and related data exchange issues regarding product data, especially libraries describing these products. A common abbreviation for ISO 13584 is PLIB - product libraries. PLIB contains a conceptual data model formally specified in the EXPRESS language of STEP (standard for the exchange of product model data).

In general, we can say that PLIB does not address web-based procurement systems and marketplaces. On the technical side, PLIB is not based on XML, because XML was not available in the mid-1990s when the PLIB development began. While the current version stems from 1998, some projects intended to add an XML-based exchange format (e.g., Pierra et al., 2000), but these development have not been included in the standard itself.

Contrary to its aim, the adoption of PLIB is quite low, especially in electronic procurement. If we look at relevant standard PCS, no system provides PLIB compliant data files.

4.2 Extended Specification

The basic components of PCS are classes and attributes. While keeping this in mind, one could assume that such a specification is only complex in terms of its number of classes and attributes, but the specification itself would be rather simple. For instance, a class definition would consist of identification, class name, long description and a reference to the super class. An attribute definition would contain identification, attribute name and description plus data type and domain of values. Mapping between classes and attributes would be the third component. But this simple model does not match the actual complexity. There are many more concepts that require both new specification elements and relationships as well.

Extended specifications, as they are introduced by standard PCS, are driven by new requirements arising from web-based procurement and sales systems. These requirements can be grouped as follows:

- The specification has to go *beyond classes* and provide detailed information about class-specific attributes, because standardized sets of attributes are a cornerstone of aligned product descriptions and efficient product comparisons. Therefore, modeling of attributes is essential for many standard PCS.
- The specification also must include *definitions of synonyms, units and values* as additional components. For instance, class synonyms in eCI@ss are defined separately from the classes. Each class synonym is defined by its identifier, name, description, version information and so on. Modeling of unites of measurement is a new concept contrary to giving references to units already defined in the Système International d'Unités (standardized in ISO 10303-41). Modeling of attribute values is necessary for customized enumerations.
- The specification has to provide *meta information* about the PCS, its structure and basic concepts. One argument for providing this meta information is to enable software systems to build the class hierarchy correctly and efficiently. This would require that there are data elements on the system level that say, for instance, "Number of Levels=4"; "Balanced tree=yes" meaning that all sub-trees have the same number of levels, e.g., any class on the third level has at least one sub-class, fourth level. Poly-hierarchies would allow that a class has two or more father classes.

If we look at standard PCS, we have to state that extended specifications result in complex data models and exchange formats as well. An analysis of the newest version 5.0 of eCI@ss underlines this (eCI@ss e.V., 2003). eCI@ss is a horizontal PCS being developed by a consortium of mainly German companies since the late 1990s. It has gained a significant relevance for e-procurement in many European countries. eCI@ss provides more than 24,000 classes, 33,450 class synonyms, and 3,667 attributes. The eCI@ss data model includes definitions of classes (16 data elements), synonyms (8 data elements), attributes (23 data elements), values (11 data elements) and relationships between classes/attributes and attributes/values.

Comparing eCI@ss 5.0 and ISO 13584 leads to the following results. Four of six data elements introduced in version 5.0 were directly adopted from ISO 13584 (e.g., definition, note, remark). The data element "coded name" is not covered by ISO 13584 at all. The reason for this element is that the class identifier itself says nothing about the position of the class in the hierarchy, while the coded name is derived from the hierarchy (for instance: 03-12-23; first level 03, second level 12, third level 23). The identifier does

not change over time, even when the class is moved in the hierarchy. The coded name can be used for presentation purposes, since it is self describing. Four data elements transfer redundant information that only can be derived from the other data. For instance, two flags indicate if there are synonyms and attributes for this class.

The definition of attributes in Version 5.0 reveals a similar picture. The number of data elements has increased from 12 to 23, four data elements were adopted from ISO 13584, several elements transfer redundant information, and some elements are not covered by ISO 13584. The same is true for synonyms and value definitions. In summary, we observe that eCI@ss is aware of ISO 13584 and adopts it. However, some eCI@ss concepts are not compliant with ISO 13584 (e.g., coded names, separate definitions of synonyms and values).

4.3 Semantics

Extended specifications add semantics to PCS. This semantics describes classes and attributes in more details, hence suppliers and all companies that create catalogs and classify products according to a given standard PCS can use these semantics in order to classify product correctly. From this view, an extended specification helps to understand the structure and content of a standard PCS. Contrary to the more detailed class definitions, the relationships between classes are not specified. All relationships between classes are “is_part_of” relationships only. This is sufficient to build a hierarchy of classes, but it does not provide additional semantics, especially no information about the scope of sub and super classes. In addition, there is no formal description what is the rationale to build a specific hierarchy. For instance, what are the requirements for a product and its attributes if it belongs to a given class?

We can provide two reasons for the lack of semantics concerning class relationships. First, the formal specification languages for standard PCS used today are not able to express these semantics. This is especially true for file formats based on CSV, XLS or MDB. The use of XML, and even more RDF (resource description framework) would enhance the semantic richness of PCS (Brickley/Guha, 2004). Contrary to the capabilities of RDF, no standard PCS provides specifications based on this language.

Second, and this reason is critical, we can assume that organizations that develop and maintain PCS do not follow strict rationales when building class hierarchies, since hierarchies and sets of classes are the result of standardization processes and require consensus between involved parties. Therefore, standard PCS concentrate on building *accepted* hierarchies rather than on defining exactly why a class hierarchy is structured as it is.

5. IMPACT OF STANDARDIZATION TRENDS ON AUTOMATING PRODUCT CLASSIFICATION

5.1 Impact on Information Retrieval

Our analysis of standardization issues has identified several standardization trends valid for standard PCS. Further, we discussed obstacles for proliferation of standards like ISO 13584, and stressed that heterogeneity that is still evident.

All IR based classifiers implement a wrapper to be able to import different PCS. The reason is that a common data model for the content of PCS is still lacking; at least the standard PCS do not adopt ISO 13584 fully, and use custom and therefore proprietary exchange formats. If standard PCS would agree on a common data model and use the same exchange format, then the wrappers would become obsolete. Importing standard PCS would benefit from this standardization trend. Implementing IR based classifiers could concentrate on IR algorithms and the integration of domain specific requirements.

Extended specifications of PCS result in more complex specification and therefore in new attributes (attribute in terms of classification, not product attributes) that have to be considered by classification algorithms. This is two-fold: on one hand, the number of attributes is rising as standard PCS add new data elements. For instance, the verbal description of a class in eCI@ss is distributed on four data elements:

preferred name, definition, note, and remark. Each data element may contain relevant attributes, thus terms must be extracted from these fields. On the other hand, the more complex attribute space delivers additional information about classes. This information contributes to the classifying process and can result in greater classification success.

The next steps to semantic richness are relationships between classes. We have seen that the degree of semantics concerning class relationships and the class hierarchy itself is very low. Additional semantics would allow improved or new classification approaches based on this formal semantic. However, this is beyond the scope of classic information retrieval (for ontology-based integration, e.g., Corcho/Gómez-Pérez, 2001).

5.2 Impact on Schema Integration

Similar to IR approaches, schema integration has to deal with the formal specification of two or more PCS that must be integrated. A common instrument to overcome schema heterogeneity is to map the given PCS data model to an internal schema. Therefore, when integrating two PCS the first step is to import the specifications. The benefit of standardization of data models and exchange formats is obvious: if standard PCS provide their specifications based on the same data model and use the same exchange format, then importing PCS data is a rather simple task. In addition, this kind of reference data model can also build the foundation for the internal schema of the integration tool. Our conclusion is that designing a tool for product classification integration is very similar to the development of a reference model describing the structure of PCS. For instance, ISO 13584 can serve as a starting point for such a reference model, and for the internal schema of the integration tool as well. If standard PCS adopt ISO 13584 partially or completely, product classification integration is also facilitated.

Extended specifications also must be considered in schema integration approaches. Two PDC may cover these extensions differently, but all extensions serve as input data for the integration problem and go into the mapping process. In general, we can assume that an extended specification provide more semantics about classes and attributes, hence this information is able to make the mapping process easier, or more successful. Since attribute usage thrives (Ondracek/Sander, 2003), mapping of different attributes will become more important, thus the formal specification of attributes is also important for the integration task.

A richer specification of relationships between classes using relationships types, formally expressed rationales, and class-specific requirements for attribute values can be seen as an additional input to the integration problem. In addition, mappings between two PCS already provided by a PCS fulfill a similar role. For instance, eOTD aims at defining a set of classes and attributes suitable for eCI@ss, UNSPSC and other PCS (ECCMA, 2003).

6. CONCLUSIONS

In this paper we have analyzed major standardization trends in product classification and evaluated their impact on automating product classification. We showed that the classification problem has to cope with heterogeneity of data models, exchange formats, granularity of formal specification, and semantic richness. Standardization addresses the need for a set of basic concepts accepted by all PCS and the organizations that develop and maintain PCS respectively. A starting point for such a standard may be ISO 13584, but it does not fulfill all requirements so far, and its usage is still low. Yet we have seen that eCI@ss 5.0 – as one of the important horizontal PCS – has adopted some concepts of ISO 13584 recently.

The results of our analysis back up recent initiatives by standardization bodies and the organizations that develop PCS as well. For instance, CEN/ISSS as an ICT standardization body in Europe (European Committee for Standardization / Information Society Standardization System) has started a project on harmonizing different data models and exchange formats for PCS (CEN/ISSS, 2004). However, new requirements from e-procurement and e-sales as well as countless initiatives to develop and improve standard PCS for different branches of industry and different markets worldwide call for independent, medium-term research and standardization work in the future.

REFERENCES

- Ariba, 2004. *cXML 1.2.011*. URL: <http://www.cxml.org>.
- Bergamaschi, S. et al., 2002. Product Classification Integration for E-commerce. *Proceedings of the 2nd International Workshop on Electronic Business Hubs*. Aix-en-Provence, France, pp. 861-867.
- Brickley, D. and Guha, R. V., 2004. *RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation 10 February 2004*. URL: <http://www.w3c.org/TR/rdf-schema>.
- CEN/ISSS, 2004. *CEN/ISSS Global Multilingual Product Description and Classification for eCommerce and eBusiness*. URL: <http://www.cenorm.be/iss>.
- CommerceOne, 2003. *xCBL 4.0*. URL: <http://www.xcbl.org>.
- Corcho, O. and Gómez-Pérez, A., 2001. Solving Integration Problems of E-Commerce Standards and Initiatives through Ontological Mappings. *Proceedings of IJCAI 2001 Workshop on E-Business & the Intelligent Web*. Seattle, USA, pp. 131-140.
- Ding, Y. et al., 2002. GoldenBullet: Automated Classification of Product Data in E-commerce. *Proceedings of the 5th International Conference on Business Information Systems*. Posen, Poland.
- ECCMA, 2003. *eOTD – ECCMA Open Technical Dictionary*. Newsletter August 4. URL: <http://www.eccma.org/eotd>.
- eCl@ss e.V., 2003. *eCl@ss Version 5.0*. URL: <http://www.eclass-online.com>.
- Fensel, D. et al., 2001. Product Data Integration in B2B E-commerce. In *IEEE Intelligent Systems*, Vol. 16, No. 4, pp. 54-59.
- ISO, 1998, *ISO/IS 13584-42 Parts Library: Description Methodology: Methodology for structuring parts families*. Geneva, Switzerland.
- Leukel, J. et al., 2002. A Modeling Approach for Product Classification Systems. *Proceedings of the 2nd International Workshop on Electronic Business Hubs*. Aix-en-Provence, France, pp. 868-874.
- Ondracek, N. and Sander, S., 2003. Concepts and Benefits of the German ISO 13584-compliant online dictionary www.DINsml.net. In *Proceedings of the 10th ISPE International Conference on Concurrent Engineering*, Vol. Enhanced Interoperable Systems, Madeira, Portugal, pp. 255-262.
- Open Applications Group, 2002. *Open Applications Group Integration Specification, Release 8.0*. URL: <http://www.openapplications.org>.
- Pierra, G. et al., 2000. From digital libraries to electronic catalogues for engineering and manufacturing. *International Journal of Computer Applications in Technology*, Special Issue on Applications in Industry of Product and Process Modelling Using Standards, Vol. 18, No. 1, pp. 27-42.
- Schmitz, V. et al., 2001. *Specification BMEcat, Version 1.2*. URL: <http://www.bmecat.org>.
- Schulten, E. et al., 2001. The e-commerce product classification challenge. *IEEE Intelligent Systems*, Vol. 16, No. 4, pp. 86-89.