Reclassification of Electronic Product Catalogs: The "Apricot" Approach and Its Evaluation Results

Sven Abels and Axel Hahn
Business Information Systems, University of Oldenburg, Oldenburg, Germany

abels@wi-ol.de hahn@wi-ol.de

Abstract

Electronic Product Catalogs (EPCs) are becoming more and more important as businesses interact electronically with one another and with customers. EPCs are the databases in which businesses store information about their products. EPCs allow customers to locate items they wish to purchase and business partners to access a business's offerings. Typically each business's EPC is organized to meet the requirements of one of many competing standards. Problems arise when various business partners use different standards to organize their EPC. Translating a product catalog from one standard to another manually is no easy task, even for a single item, and the typical EPC contains thousands of items. This situation is known as the reclassification problem. The paper describes the problem in greater details and also proposes a solution, which we dub "the Apricot approach".

The article starts with a brief overview about EPCs and classification systems. It then provides a description of the reclassification problem and describes existing solutions. Next, the Apricot-approach and its implementation are described. This article provides evidence that the Apricot Approach is useful and fruitful.

Keywords: Classification, Reclassification, Electronic Product Catalogs, Apricot, eCl@ss, UNSPSC, e-Business, Product Ontologies.

Introduction

In recent years, e-Business has gained attention, and buying and sellings products online in an important component of e-Business. Today for many companies its e-Business value is much higher than the traditional way of conducting business. E-Business enables shop owners to offer a huge number of different products from various suppliers. Electronic Product Catalogs (EPC) enable customers to locate and order products. EPCs started enhancing or, in many cases, even replacing the traditional area of paper-based catalogs. They provide a shop owner a cost effective way of presenting products to the customer without being bound by constraints of paper-based catalogs, such as a maximum number of pages. EPCs can inform clients when product data are updated or added. The larger number of potential products is, of course, beneficial for (i) customers, who have more choices of products, (ii)

Editor: Eli Cohen

shop owners, who can keep product data updated, and (iii) manufacturers, who are able to reach better product placing in online shops. In some scenarios, a shop owner is sent a catalog from each manufacturer containing all available products. In many cases, these catalogs contain thousands of products. A study conducted by Abels and Hahn (2006) found the number of products per catalog averaged about 34,000.

Although having that many products in catalogs has many advantages, the large amount of product data raises new problems. One major problem is providing an overview about the products and their attributes. This is where so-called classification systems come. Classification systems in e-Business have been developed, "to assign each product to a product group corresponding to common attributes or application areas" (Leukel, Schmitz & Dorloff, 2002). Classification systems will be introduced in details in the following subsections together with product catalogs.Basically they aim in providing a common structure for grouping similar products.

After giving a brief introduction about classification systems and electronic product catalogs, the next section describes today's problems with classification systems. This article focuses on the reclassification of product data. It gives an overview about existing approaches and introduces our Apricot framework for reclassifying products. This approach helps product suppliers inform their clients about new or updated product data by sending a product catalog in the specific classification system of the client. We demonstrate the functionality of Apricot and present evaluation results that have recently been finished. The novelty of Apricot is the aproach itself (unlike other solutions, it interprets existing classification information) as well as its application to e-business catalogs in terms of using and evaluating the tool.

The purpose of this article is to:

- give an overview about the meaning of classification systems as a key factor in informing clients,
- explore the problems of classification systems in practice,
- focus on the reclassification problem and provide an overview about possible solutions,
- introduce the Apricot framework for reclassification and
- describe the success of this approach in real-world scenarios, providing an overview about the evaluation results of the prototype.

Electronic Product Catalogs

Electronic product catalogs are used to store various product data in a homogenous catalog, which is often divided into different category groups. Baron, Shaw & Bailey (2000) define an electronic product catalog as "electronic representations of information about the products and/or services of an organization." Muldoon (2000) divides those catalogs into catalogs targeting the end consumer, called the B2C-catalog, and catalogs targeting other business-partners (B2B). He emphasizes that both are important for modern e-Business.

Since several years ago, various different formats are available for storing product catalogs. Some of them are based on XML and provide a high flexibility for different product types and different requirements. Examples for common product catalogs are BMEcat, cXML, OAGIS or XCBL. A comparison of those formats and their functionality and purposes is described in Quantz and Wichmann (2003). The choice of a catalog format depends on the application area as well as on the functionalities needed. Moreover, it depends on requirements of business partners and existing software solutions. It is important to consider the version number of a standard used. For example, the catalog formats BMEcat and OAGIS provide many important enhancements in their current version that were not contained in earlier versions. Though both formats are downwards compatible, which in this case means that they allow the usage of files based on a new version in

systems that expect files based on the old version and vice versa, the full potential can only be revealed if both partners are using the same version of a standard.

For this purpose it necessary that both the supplier of product data and their consumer; for example, an operator of a web shop or the user of an e-procurement application, agree on using a common format to exchange product data. The usage of a standard format enables an exchange of different product data with multiple business partners. This makes it easy to integrate the data of new business partners. Dorloff, Schmitz and Leukel (2002) describe advantages that standardization in this area can have and they reference Olson (2000), who described differences of catalog exchanges in B2B domains, compared to B2C environments. He states that (i) the interaction between information systems is essential, that (ii) the business content is diverse and complex, and that (iii) the control mechanism ranges from one-sided to peer-to-peer relationships. Dorloff, Schmitz and Leukel therefore suggest fostering the usage of standard formats in this area.

Whenever different e-Commerce partners are using different catalog formats, a transformation from one format into another one is needed. Marron, Lausen & Weber (2003) demonstrate this by using XPath (see also Clark & DeRose (1999) for a similar application). Other transformation approaches can be found in Omelayenko and Fensel (2001) or in Poulovassilis and McBrien (1998).

Classification Systems

As mentioned at the beginning of this section, classification systems are used to assign each product to a specific category. In classification systems, those categories are called classes (see DIN 32705, 1987).

Classification systems can either be defined within the product catalog itself or they can be defined externally. Those classification systems that are defined within the catalog itself are catalog specific, and they tend to change from catalog to catalog. They are often called product groups. Most of the catalog formats available today provide the possibility to define product groups and to arrange products into them. In cases where the classification system is defined externally, the products are assigned to a class by using a class-code. For example, the string *49-23-15-13* in product descriptions based on the UNSPSC-System (http://www.unspsc.com), defines that this product is a toy train. Classification information can easily be integrated in many modern catalog

```
...
<Article>
   <Supplier AID>57663</SupplierID>
   <Article_Details>
      <Description_Short>XtraTrain 2005</Description_Short>
      <Description_Long>
            A battery driven toy train with blue lights for every age
      </Description_Long>
...
   <Article_Price>
        <Price_Amount>45,50</Price_Amount>
        <Price_Curency>USD</Price_Currency>
    </Article_Price>
...
<Article_Features>
 <Reference_Feature_System_Name>UNSPSC-5.0</Reference_Feature_System_Name>
 <Reference_Feature_Group_ID> 49231513</Reference_Feature_Group_ID>
</Article_Features>
...
```

Figure 1: Product catalog fragment with a product assigned to 49231513

formats, such as BMEcat, without any problems (see Hentrich, 2001). Figure 1 shows a small fragment of a BMEcat catalog showing a product that is assigned to the class *49-23-15-13*.

A classification system consists of an unchangeable list of classes (see Grabowski, Lossack & Weißkopf, 2002). Almost all classification systems currently used to classify products are based on a hierarchy. In order to complement this, classification systems sometimes provide a set of keywords, descriptors, and attributes for each class. Well-known product classification systems are, for example, UNSPSC, ETIM or eCl@ss (see http://www.etim.de and http://www.eclass-online.com). Furthermore, the Technical Dictionary of RosettaNet (RNTD), a subsidiary of the Uniform Code Council (UCC), is often referred to in this domain (see RosettaNet, 2004). In the literature, other ordered structures are often called a classification system too, such as the order structure of eBay. A comparison between those three classification systems can be found in Beneventano, Guerra, Magnani, and Vincini, (2004). At the top of Figure 2(a) an example for a structure of a classification system is shown. As seen in the figure, classification systems may define a set of attributes for each class. Those attributes are used to specify the properties of products located in a class. For example, they can specify the shape or the color of a product. Usually, those attributes are inherited along the classification hierarchy. Products assigned to a class can, therefore, define values of all attributes of the class.

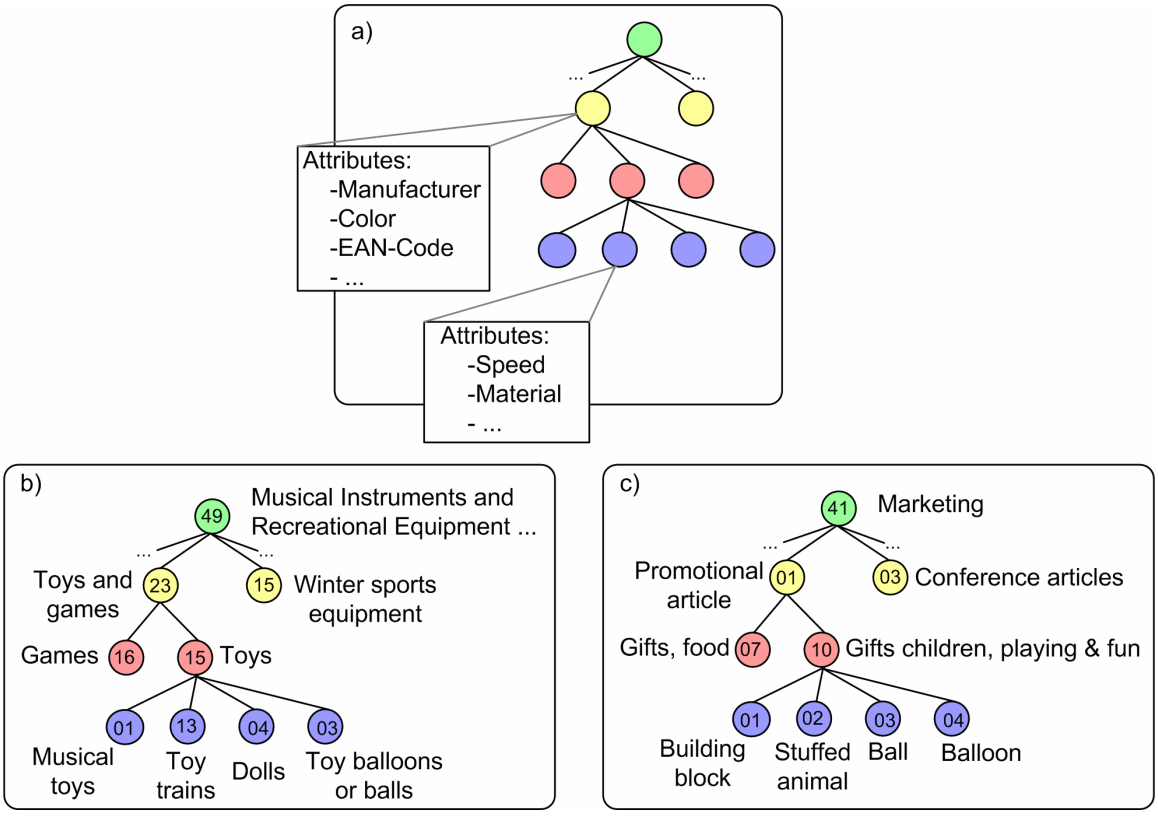

Figure 2: Structure of classification systems (a) and two examples (b), (c)

At the bottom of Figure 2, fragments of two classification systems can be seen in (b) and (c). (Their attributes are not displayed because of space limitations.) The first one (b) shows a fragment of UNSPSC 5.0, while the second one shows eCl@ss 5.0. As seen in the figures, the actual code of a specific class is evolved from the codes of its upper classes. For example, the class "toy trains" is 49-23-15-13, which is the concatenation of all upper classes (49-23-15) plus its own code (13). This makes it easier to quickly sort products or to group them based on different levels.

Advantages for clients when using classification systems

Classification systems offer many advantages to customers dealing with product catalogs. The most important advantages are the possibility of grouping similar products. This enables customers to easily find similar products even though they have different product descriptions. Another advantage is that classification codes are independent from the natural language of the product description. This enables clients to search for products in all languages. Compared to simple keyword searches, a client does not have to know the exact keyword to find a product. For example, searching for "notebook" will normally not list products that are described with the word "laptop" when relying on a keyword search. There has been a lot of discussion about using classification systems and their advantages in literature within the last years. For further discussions, the following publications can be recommended: Dorloff, Schmitz, and Leukel (2002), Ding et al. (2002) and Abels and Hahn (2006).

Main Tasks and Problems

There are many different classification systems that can be used commercially without cost. In many cases, they are maintained and updated by a consortium that adds classes, and changes or removes them during the years. When applying classification systems to real-world product catalogs, then two main tasks can be identified: classification and reclassification (see Fensel et al., 2001).

Classification

The first task is the initial classification of product data. When using a classification system, it is necessary to assign each product of a catalog to a specific class of the classification system. It is therefore necessary to analyze the product and to find a corresponding class for this product. This task can be performed manually or in an automatic or semi-automatic approach.

Classification systems in current use tend to have many different classes. For example, eCl@ss 5.1 contains more than 25.000 different classes completed by more than 21.000 keywords. UNSPSC 7.0 defines more than 18.000 different classes arranged in a hierarchy with 55 classes on the first hierarchy level. Hence, companies that use classification systems usually have to deal with a large number of classes, which sometimes makes it hard to clearly identify the correct class for a product and which leads to a relatively high amount of time needed for classifying products. In Grabowski et al. (2002) the authors identify about 5-10 minutes of work for each product when using a manual classification. The reason is the high number of similar classes in most classification systems. Using an automatic or semi-automatic process can help to speed up this process. For example, a semi-automatic classification of product data is provided by Storeserver Classifier (see Storeserver, 2005) or e-proCAT (see e-proCAT, 2005).

Reclassification

The second important task that appears when using classification systems is the reclassification of product data. According to Beneventano and Magnani (2004) this is the process of classifying a product that has already been classified before. This becomes necessary whenever a product catalog has been classified using a classification system that is not identical to the classification system needed for further processing. For example, one might have used UNSPSC to classify a product catalog but the business partner might use eCl@ss. In those cases, it is necessary to switch from one classification system to another one. Hence, all products of the catalog have to be assigned to the new class structure. As stated by Tanenbaum (1996), "the nice thing about standards is that there are so many of them to choose from".

The client (e.g. a small company) of a product supplier often uses an e-procurement system to exchange product data. Clients can use product catalogs of suppliers to integrate products into their e-procurement system, allowing them to order products electronically. If client and supplier are using different classification systems, then an integration of the product catalog into their own classification structure within the e-procurement system requires manual work. Providing an automatic or semi-automatic reclassification will help to better inform clients of new product data. The reason is that the catalog of the supplier can automatically be reclassified before it is integrated into the e-procurement system. This allows the client to easily access the new catalog data in a faster and much more cost effective way.

Existing Solutions for Reclassifying Products

When looking at existing approaches and their implementations for performing an automatic re-classification of product data, two different types can be identified:

1. Approaches based on a mapping between the two classification systems.
2. Approaches based on analyzing product data (mainly descriptions)

The first type is based on the idea of creating a mapping between the two classification systems. For example, a mapping might be defined that says "49231513" in UNSPSC is the same as "41011006" in eCl@ss. Using a table of those mapping definitions enables us to simply search-and-replace the existing classification information with the new class coded. Table 1 shows an example fragment of such a table:

Table 1: Fragment of a mapping between UNSPSC and eCl@ss

Source		Destination
Unspsc:industrialpaper	→	eCl@ss:writingpaper
Unspsc:businesspaper	→	eCl@ss:writingmaterialdrawingmaterial
Unspsc:toiletpaper	→	eCl@ss:hourseofficesantcleaner
. . .	. . .	. . .

There are several approaches that describe how to create such a mapping. For example, Beneventano and Magnani (2004) and Benetti (2001) describe those approaches.

For a reclassification of product data in e-Business, the use of mappings allows a very fast re-classification with low efforts, once the mappings are produced. An exclusive usage of mappings is, however, normally not possible because, in most cases, there is no isomorphism between the systems. Ding, Fensel, Klein, Omelayenko, and Schulten (2003) mention the following problems, "Mapping the content standards by specifying pairs of equivalent categories is not always possible due to different principles used to aggregate the products into categories of the same abstraction level. For this reason, for example, mapping UNSPSC to eCl@ss includes creation of many-to-many bridges regrouping the products to categories." In Schulten et al. (2001) similar problems have been identified. It is most likely that the first classification has a different hierarchical structure and depth than the second one. Because of this, there is a lack of information for classifying a product correctly when limiting to a pure mapping approach. For example a classification B system might have a category called Paper in the main category Office Materials. B might now need an additional break down into White Paper, Recycled Paper, etc. (see Figure 3). Hence, additional information is needed to re-classify all data correctly. To perform the re-classification it is in this case not enough to map the categories of both classification systems, but to analyze each

product data independently and it is, of course, desirable to automate this classification and re-classification processes as much as possible to save both time and costs.

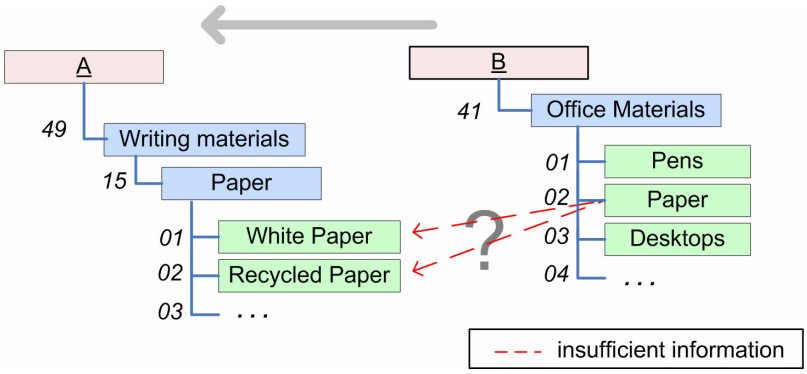

Figure 3: Problems of an application of mapping information for re-classification

Since the classes of the two classification systems are in most cases very different, the described problem appears in many mappings. In Häusler (2005) an example for 50 classes is given for a mapping between UNSPSC and eCl@ss. The result was that in the average, each eCl@ss class could be mapped to about 6 classes in UNSPSC. This means that this approach is not suitable for performing a complete reclassification of product catalogs.

The second type consists of approaches based on analyzing the data of each product in order to assign the product to an appropriate class. Those are basically classification approaches that neglect the existing classification information and that try to perform a complete new classification of the product into the new taxonomy.

Examples of existing implementations are Goldenbullet (see Ding et al, 2002), PAK (see Grabowski et al., 2002), ePro-CAT and Storeserver Classifier. Their main information source for performing the classification is the interpretation of product descriptions and keywords. The success of a classification based on such an analysis of the descriptive product data varies strongly and is dependent on the concrete approach. Ding et al. (2002) indicate that to achieve a precision of 78% with their Approach called GoldenBullet, they use a Naïve-Bayes classification in GoldenBullet to classify 40% of the products, while they used the other 60% as training data for the algorithm. They testified that this rate is already higher or at least equal to the error rate of a manual classification. In Wolin (2002), the AutoCat system is introduced, which is able to classify between 77.3% and 79.5% correctly. Agrawal and Srikant (2001) use an approach that considers the similarity between products when classifying them. They use the taxonomy of Google and Yahoo for evaluating their approach and reach between 58.9% and 64.4%. In summary, existing reclassification schemes have severe problems.

The Apricot Approach

The main task of the classification process is to find the correct class for a product. In the case of a reclassification process, we have the same task but this time we have additional information that can be interpreted: this is the information of existing class assignments of the product. This information is neglected in most existing solutions since they were not developed for the reclassification but for the classification of products. In our Apricot approach, we use this information to perform a higher success rate than most other approaches. Apricot is an integrated approach that is, on the one hand, able to interpret product data such as the product description and, on the other hand, considers existing classification information as well. In this section, we describe the Apricot approach to provide an overview about how Apricot works.

Our approach can be divided into several interconnected phases that are arranged as a circle in order to reclassify all product data. In our approach all products are reclassified sequentially. The phases of Apricot can be summarized as:

1. Preparation of language based and catalog based information,
2. Cleaning & Enhancement,
3. Analysis,
4. Combination,
5. Class selection & attribute generation, and
6. Enrichment.

Figure 4 shows these phases and their succession. As soon as the reclassification process has been finished, the product data are stored in an output catalog and the Apricot approach is finished.

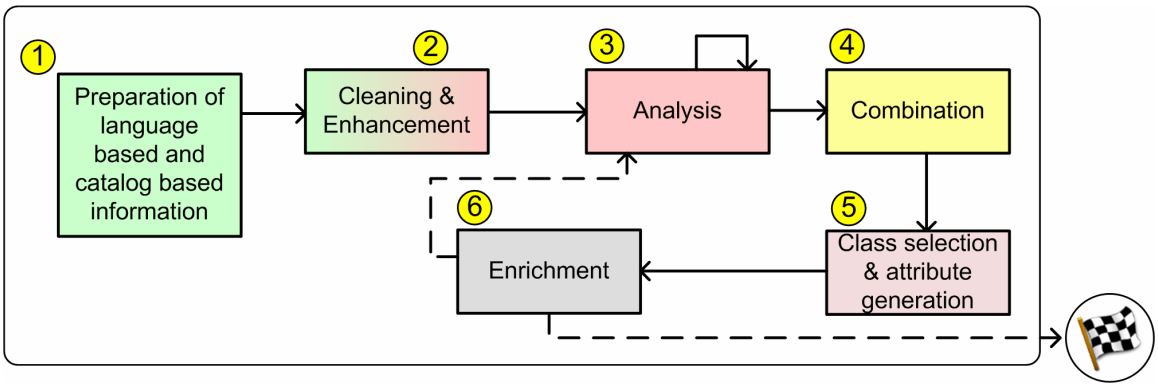

Figure 4: Different phases of the Apricot approach and their succession

Phase 1: Preparation of Language Based and Catalog Based Information

The preparation phase is used to import all necessary information needed for performing the reclassification process. This includes three different kinds of information:

1. Information from the product catalog.

   The product catalog itself is parsed and imported into a data structure that represents the product catalog, including all of its information. For the implementation of Apricot, we realized the support of BMEcat product catalogs. Hence, an XML-parser for BMEcat is used in this phase to parse the catalog and to retrieve all product information, such as product descriptions, price information and product groups.

2. Information from the specifications of the classification systems.

   Apart from the catalog information, the class hierarchy of all involved classification systems is needed. Hence, their class structure is imported and represented in a data structure within Apricot. It is important to not only import the class names but to also consider class attributes, keywords and synonyms if available.

3. Language information used in the later phases.

   Among other data, class names, product names and attribute names have to be interpreted in a later phase of Apricot. Because of this, it is necessary to import some language dependent information such as a dictionary or a thesaurus. In order to assure that the ap-

proach is completely language independent, that information is imported based on the language of the catalog that has to be reclassified.

Phase 2: Data Cleaning & Enhancement

To detect similar classes and to interpret product related data, it is necessary to adjust all data to each other. For example, a price in USD for a product could be written as "12,34 $" or as "USD 12.34" or it might even be available in a completely different currency. Apricot tries to recognize this and to adjust all data to a common format with identical units in phase 2.

Moreover, text information is cleaned, too. This is done by using a language based stemming algorithm, such as the Porter Stemmer (see Porter, 1980). This makes it possible to replace all words with their basic form. For example, the word "houses" is replaced with its singular ("house"). Furthermore, so-called stopwords are removed. Stopwords are words that do not have any influence for the further processing. Examples in the English language are "he", "the", "for", etc. Further information can be found in Heyer, Quasthoff & Wolff (2002).

Apart from adjusting information, this phase performs an enhancement of the existing information. This is done by (a) performing a word splitting of compounds ("football" → "foot" + "ball") and by (b) adding synonyms wherever possible ("laptop" → "notebook").

Phase 3: Analysis of Data

The third phase is the main phase of the approach that integrates several different approaches, called "analyzers," for interpreting the data from the product catalog and from the existing classification information. An analyzer might, for example, be based on an approach that uses the product keywords as a base for assigning a product to a class by comparing the class name with the product keywords. It might also be a machine learning approach using the product descriptions as a base for finding possible class assignments.

A controller forwards the product data to all analyzers. They can be divided into three different types:

1. **Classifiers:** Classifiers are based on the product catalog data. They analyze product properties in order to find potential classes for a product. For example, they might analyze the product description or its keywords and compare it with the class names of the classification system.
2. **Reclassifiers:** Reclassifiers are based on the existing classification information. They try to look at the current position of the product in the classification systems. For example, a product might already be classified based on UNSPSC and it might furthermore be located in a product group. A Reclassifier uses this information to find possible classes in the structure of the new classification system.
3. **CategoryFilters:** While Classifiers and Reclassifiers both result in a list of potential classes for the product, CategoryFilters use another approach. They analyze all classes in order to remove those filter classes that are very unlikely for the product. For example, if we know that a product costs about 5,- $ then it is very unlikely that this product belongs to a class that consists of 1000 other products each costing more than 500,- $. Our evaluation of Apricot has shown that the analysis of product prices is, for example, able to filter about 7.73% of all classes.

Figure 5 shows the analysis phase. Each analyzer contains a priority that can be changed by the user. This makes it possible to change the influence of an analyzer if necessary. For example, one

might want to define that an interpretation of product descriptions is more significant than the interpretation of existing classification information.

After applying all analyzers, their results are collected and forwarded to the next phase of Apricot that deals with the combination of the results.

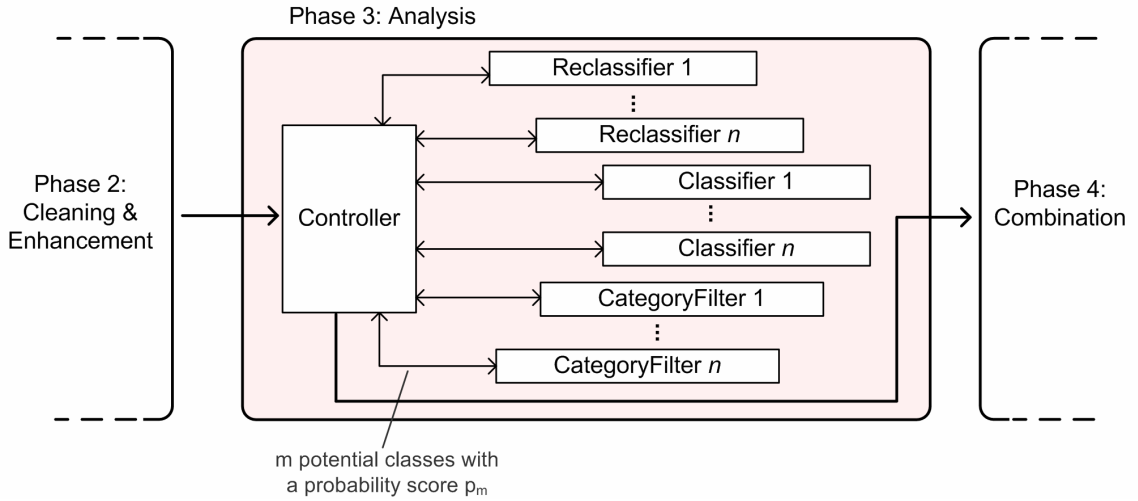

Figure 5: The third phase consisting of a controller and a set of analyzers

Phase 4: Combination

Within the combination phase, Apricot combines the results of all analyzers based on their priority. This phase aims at creating an aggregated list of potential classes for the product. This list is ordered by a score value that represents the probability of the class belonging to the product. For example, Table 2 shows on the left part the input of the combination phase. This input consists of a list of potential classes and its probability for each analyzer as well as a priority value. The right part of the table shows the output of this phase. In Table 2, we have shown a simple aggregation that uses the priority as a weight factor. In this case, the priority was chosen manually. Within the Apricot framework, it is possible to change the way, figures are combined. There is also an approach for automatically determining the priority of all analyzers. For more information, please look at Abels and Hahn (2005).

Table 2: Input and output of the combination phase

Results of analyzers					**Aggregated results**	
Analyzer-No	Priority	Class	p		Class	Score
1	3	Class_1	0,50		Class_1	6,00
1	3	Class_2	0,25		Class_2	0,75
2	6	Class_1	0,75		Class_3	1,50
2	6	Class_3	0,25		…	…
…	…	…	…			

Phase 5: Class Selection and Attribute Generation

The fifth phase of Apricot, class selection and attribute generation, consists of two sub phases. The first one is the selection of a class for the product. This can either be an automatic or a manual step. In an automatic approach, the class with the highest score in the table that has been produced in the last step is chosen. In case of using a manual selection, the user is involved. For each

product, the product data and the first entries of the table created in phase four are displayed to the user. The user is then free to select the correct class for each product based on the suggestions of Apricot.

Once the class has been selected, Apricot generates attributes for the product based on the attribute list of the class. As stated in the last section of this article, some classification systems, such as eCl@ss, provide a list of attributes for each class. Those attributes are, for example, the manufacturer or the color of the product. Apricot tries to generate these attributes automatically. This is done by using three information sources:

1. Existing attributes from other class assignments.

   When reclassifying products, we assume that the product has been assigned to at least one other class structure before we classify it to the classification system that we need. For example, we might need the product data to be classified based on eCl@ss and it might have been classified based on UNSPSC and ETIM before. It is also possible that the product was classified before based on the same classification system in a different version than the one that we need. For example, we might need eCl@ss 5.0 and the product might already have been classified based on eCl@ss 4.1. -- Hence, there are a lot of cases where the class attributes were already defined in the existing classification information. Apricot, therefore, looks at this information and tries to find identical or very similar attributes. If Apricot finds such an attribute, its value can be reused to generate the new attribute.

2. Product properties from the catalog data.

   Several class attributes such as the product manufacturer or the EAN number of a product, might also be specified as product properties within the electronic product catalog data itself. For example, BMEcat catalogs contain an extra field for specifying the manufacturer of each product. In those cases, Apricot uses the product data to generate the attributes, as needed by the class.

3. Extraction of information from the text description of the product.

   In cases where Apricot was not able to find the attribute data in existing classification information or in the product catalog data, it is possible to scan test-based product information, such as the product description, for finding out the attribute values. (This method is not yet included in the current implementation of the approach) This is, of course, a very vague way of detecting product attributes but it can be useful for some specific attributes. For example, if we find a string *10” x 12” x 80”* in the product description then it is very likely that these values represent the dimensions of the product.

At the end of the attribute generation, the suggested values are displayed to the user so that he or she can change or complete the data if necessary.

Phase 6: Enrichment

At the end of the last phase, the product is classified completely. In the sixth phase, the class assignment is used to enrich the analyzers. Since some analyzers might depend on a machine learning approach, the assignment is forwarded to a special interface of the analyzers allowing them to use the decisions as a base for future assignments.

Once, the enrichment is finished, Apricot either continues with the next product or it ends the process by storing the reclassified catalog with all necessary information.

Realization and Evaluation Results

Apricot was conceptualized and implemented in a two years project. Its first prototype and the evaluation of this prototype have just been finished allowing us to present the evaluation results in this section.

Realization

Apricot was realized as a framework for automatic or semi-automatic reclassification and therefore it does not have a specific user interface. Instead, it provides methods for other software applications to start and manage the reclassification process. To evaluate the framework and to gain an impression of its capability, we created a demo application in Java. This application can be used to completely reclassify BMEcat based product catalogs into several classification systems. In our first prototype, we integrated eCl@ss 3.0, 4.0, 4.1 and 5.0 as well as ETIM 2.0 and UNSPSC 5.0 and 7.0. Figure 6 shows a screenshot of Apricot that displays the classification window allowing the user to select a class for a product during phase 5 (class selection).

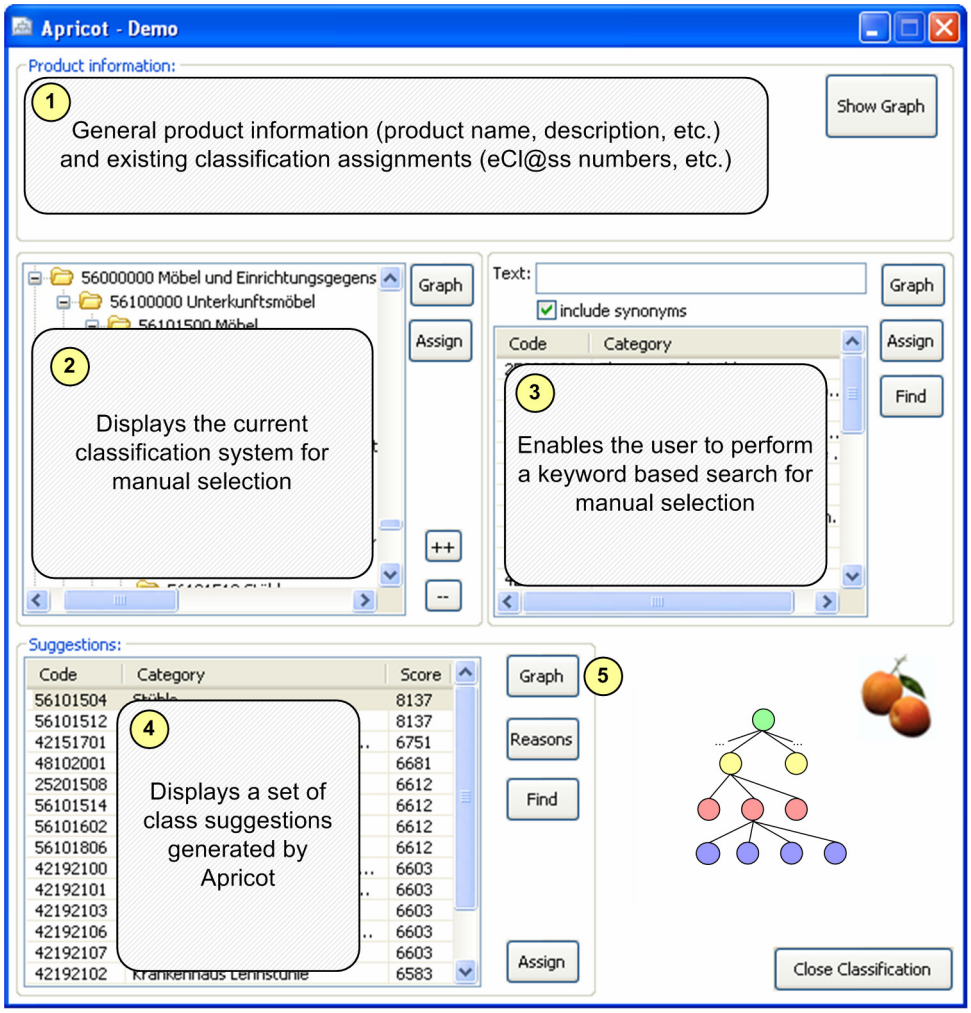

Figure 6: Screenshot of the Apricot demo application

As seen in Figure 6, the window displays some product information (1) and it offers the user to either manually select a class of the classification system or to select a class from the suggestions

of Apricot (4). In case of manually selecting a class, the user can either use the class tree of the classification system (2) or can search classes by keywords and their synonyms (3).

The Apricot demo software monitors the user's behavior in order to collect statistical data. We collect the following information:

- Percent number in which the user used the first suggestion of Apricot (Top-1)
- Percent number in which the user used one of the first 3 suggestions of Apricot (Top-3)
- Percent number in which the user used any suggestion of Apricot (Top-100) (We limited the number of suggestions from Apricot to a maximum of 100 classes)

Clicking on the "Graph" button of the window (5) displays an interconnected graph of information used by Apricot to reclassify the product. It opens another window with graphical nodes that can be clicked and enhanced to view, for example, the product catalog and its classification information graphically. Figure 7 shows a screen shot of this information for a product "GrandStar 2005".

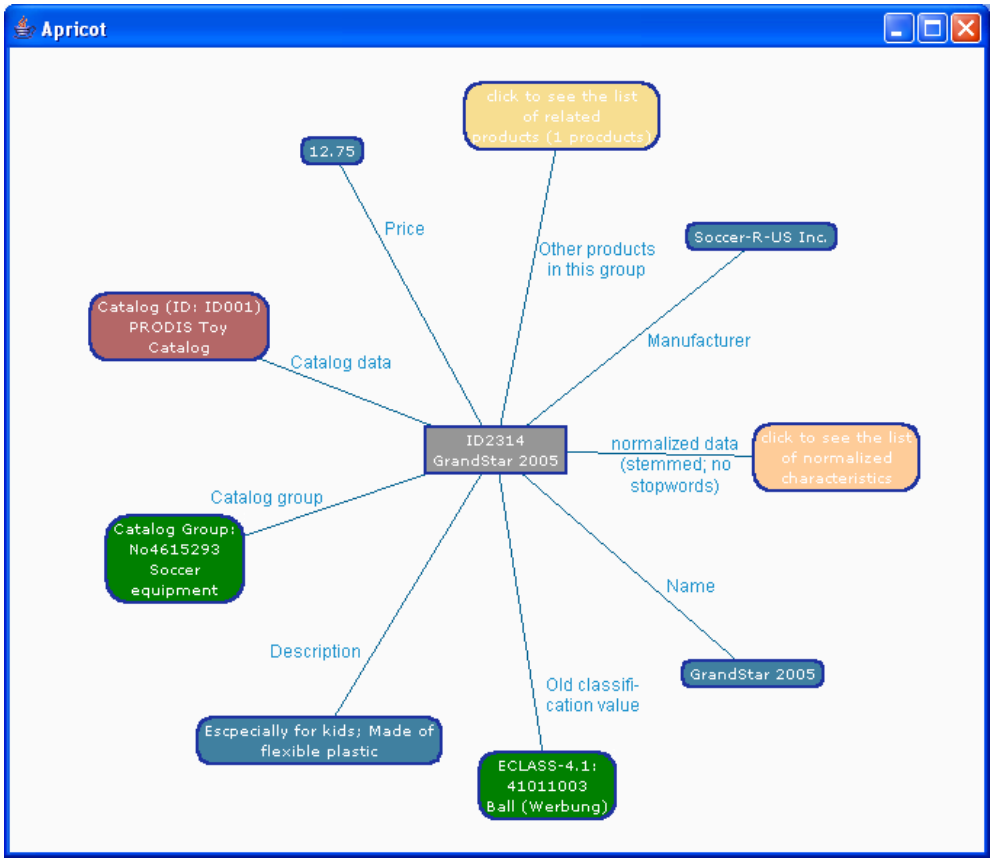

Figure 7: Interpretable data when reclassifying products

Evaluation Data and Evaluation Results

To perform an evaluation of Apricot, the demo application of the Apricot prototype was used to monitor the user's behavior as stated above. For evaluation purposes we used a test catalog based on real-world product data from the "office material" domain. We created a test set of 100 products and classified them manually based on UNSPSC 5.0. Furthermore, we arranged the products into 29 different product groups. For our evaluation, we chose eCl@ss 4.0 as our destination clas-

sification system for the reclassification process. All products in our test catalog got price information, a product name and a short product description (one sentence per product). Some of the products contained a manufacturer name and some keywords.

Duration of the reclassification process and memory needed

In our test scenario, Apricot needed between 3 and 15 seconds depending on the hardware and depending on the product details. Afterwards, Apricot displayed the suggestions to the user. Including the time each user needed for choosing a class, the whole process took about 1.25 minutes on average. Compared to a manual classification, this is between 4 and 8 times faster since manual classification usually takes between 5-10 minutes (Grabowski et al., 2002). For our test scenario, about 256MB of RAM was used by Apricot to manage the classification systems and the catalog data.

Quality of the reclassification process

As explained above, the demo application monitored the Top-1, Top-3 and Top-100 values when performing the evaluation. We distinguish between the first 50% of the product catalog and the second 50% because some of our analyzers used a machine learning approach allowing them to continuously increase their quality during the reclassification process.

We included a special analyzer that received a set of keywords defined by the user. Those keywords were used to modify the suggestions of Apricot to re-score them. The idea is that product catalogs normally belong to a specific domain. For example, they deal with office material or sports equipment, etc. Therefore, classes that deal with this domain are more likely than classes from other domains. In our approach, this set of keywords is called a “context”. We evaluated the Apricot approach using three different scenarios with (i) no definition of a context, (ii) a very short set of 3 words (short context) and (iii) a detailed context with 29 words. For example, the short context consisted of { “office”, “material”, “desktop” }.

The results of our evaluation can be found in Table 3.

Table 3: Evaluation results of the Apricot approach

Type	No context	Short context	Detailed context
1st 50% of the product catalog: (with no or only a few machine learning effects)			
Top-1	64%	38%	68%
Top-3	82%	70%	84%
Top-100	94%	92%	96%
2nd 50% of the product catalog: (with machine learning effects produced by the 1st 50%)			
Top-1	76%	46%	76%
Top-3	86%	76%	88%
Top-100	98%	94%	98%

In the first set of rows representing the first 50% of our product catalog, all machine learning analyzers are untrained. Hence, there is no machine learning effect that could be used by any analyzer at the beginning of the process. Hence, the results of the first rows can be viewed as the accuracy rates that are produced when using the Apricot approach without training. As the reclassification process goes on, the analyzers start learning from products that have already been classified correctly. Hence, the accuracy increases because of those machine learning effects. In Table 3, this can be seen by looking at the numbers of the second 50% of the catalog. Those numbers can therefore be interpreted as the accuracy of Apricot approach with training.

As seen in Table 3, in about 76% of all cases, the first suggestion of Apricot is accepted by the user and is, therefore, considered to be a correct suggestion. In about 88% of all cases, the correct class was included in the first 3 recommendations and in up to 98% of all cases the correct class was at least within the set of suggested classes of Apricot.

Surprisingly, the definition of a context, in terms of defining keywords for the catalog, is not always beneficial. The evaluation has shown that a small set of words actually decreases the quality, while a detailed set can slightly increase the overall results.

We repeated our evaluation for two external product catalogs provided by two companies that deal with office material, and we got very similar percent numbers with those catalogs. (There was only a small deviance of between 5%-10%.)

Comparing the evaluation results with existing solutions that have been introduced in an earlier section of this article, we can see that the overall results outperform many products in quality. One of the major advantages of the Apricot approach is, however, the possibility to use the approach with good results from the beginning of the product catalog. It is not necessary to provide a large amount of training data or to pre-process a certain percentage of the catalog. Apricot starts with a good success rate right from the first product and increases the quality during the reclassification process.

Since UNSPSC does not support attributes, we repeated the evaluation again by reclassifying products that were based on eCl@ss 4.0 to eCl@ss 5.0. Looking at the generation of product attributes, Apricot was able to generate about 85.7% of all attributes automatically by looking at existing classification attributes and by interpreting catalog data.

Conclusion and Further Research

The evaluation of the prototype has shown that Apricot provides a highly flexible and fast approach that is able to provide a high quality for a semi-automatic reclassification. Compared to other solutions, Apricot reuses existing classification information and it is based on the integration of several different analysis approaches. It can be used as a way for performing a semi-automatic reclassification by interacting with the user or it can be used as a way to identify a list of potential classes for a product that is used as a basis by other applications. This can speed-up the integration of new products from a supplier into the systems of his client. Hence, the exchange of information between the supplier and the client is improved. The realization and the evaluation have shown that the reuse of existing classification information can be an important information source for performing semi-automatic reclassifications.

Since Apricot is still a prototype, it would be interesting to perform some additional testing in real-world scenarios by integrating Apricot into existing product catalog systems any by using it in a long-term in order to identify weak points and additional requirements. The test catalog is currently based on two real world catalogs that have been combined. In the next phase, we want to perform a long-term evaluation by integrating our system into a real-world e-procurement system. This will give us detailed information on how the system develops within a large time scope.

This will also allow verifying the long-term development of the machine-learning analyzers. However, since it is necessary to perform this evaluation in a real-world environment with multiple users, this experiment is scheduled to last at least six months. We hope that it will help to identify parts of the system that might be improved.

References

Abels, S., & Hahn, A. (2005). Facing the product reclassification challenge. *Proceedings of the Workshop on Product-related Data in Information Systems* (PRODIS2005), INFORMATIK2005, p. 68. Heidelberg: Springer.

Abels, S., & Hahn, A. (2006). Empirical study on usage of electronic product classification systems in e-commerce organizations in Germany. *Journal of Electronic Commerce in Organizations*, *4*(1), 33-47.

Agrawal, R. & Srikant, R. (2001). On integrating catalogs. *Proceedings of the 10th International World Wide Web Conference*, Hong Kong.

Baron, P. & Shaw, M. J. & Bailey, A. D. (2000). Web-based e-catalog systems in B2B procurement. *Communications of the ACM, 43*(5), 93-100.

Benetti, I. (2001). SI-Designer: An integration framework for e-commerce. *Proceedings of the IJCAI-01 Workshop on E-Business & the Intelligent Web*.

Beneventano, D., & Magnani, S. (2004). A framework for the classification and the reclassification of electronic catalogs. *ACM Symposium on Applied Computing*.

Beneventano, D., Guerra, F., Magnani, S. & Vincini, M. (2004). A web service based framework for the semantic mapping amongst product classification schemas. *Journal of Electronic Commerce Research*, *5*(2). Retrieved from http://www.csulb.edu/web/journals/jecr/issues/20042/Paper4.pdf

Clark, J. & DeRose, S. (1999). XML Path Language (XPath) Version 1.0. Retrieved 17th Sept. 2004 from http://www.w3c.org/TR/xpath

DIN 32705 (1987). Deutsches Institut für Normung, Klassifikationssysteme; Erstellung und Weiterentwicklung von Klassifikationssystemen [Classification systems; Creation, evolution of classification systems]. *Beuth Verlag*.

Ding, Y., Fensel, D., Klein, M., Omelayenko, B. & Schulten, E. (2003). The role of ontologies in eCommerce. In: S.Stab & R. Studer (Eds.), *Handbook on ontologies*, Springer.

Ding, Y., Korotkiy, M., Omelayenko, B., Kartseva, V., Zykov, V., Klein, M. C. A., et al. (2002). GoldenBullet in a Nutshell, FLAIRS-2002: *The 15th International FLAIRS Conference* (pp. 403-407).

Dorloff, F.-D., Schmitz, V. & Leukel, J. (2002). Coordination and exchange of XML catalog data. *Proceedings of the 5th International Conference on E-Commerce Research* (ICER-5).

e-proCAT (2005). e-pro solutions GmbH. e-proCAT 3.5, *Handbook.* Retrieved from http://www.epro.de

Fensel, D., Ding, Y., Schulten, E., Omelayenko, B., Botquin, G., Brown, M. & Flett, A. (2001). Product data integration in B2B e-commerce. *IEEE Intelligent Systems* (July/August 01), S.54-S.59.

Grabowski, H., Lossack, R., & Weißkopf, J. (2002): Datenmanagement in der Produktentwicklung [Data management in product development]. *Hanser-Verlag*.

Hentrich, J. (2001) B2B-Katalog-Management. *Galileo Business.*

Heyer, G., Quasthoff, U., & Wolff, C. (2002): Möglichkeiten und Verfahren zur automatischen Gewinnung von Fachbegriffen aus Texten [Possibilities and approaches for extracting terms out of texts]. *Proceedings of the Innovation forum Content Management - Digitale Inhalte als Bausteine einer vernetzten Welt*, 2002

Häusler, S. (2005). Mapping zwischen Klassifizierungsstandards im E-Business [Mapping between classification standards in e-Business]. Bachelor Thesis, University of Oldenburg.

Leukel, J., Schmitz, V. & Dorloff, F.-D. (2002): Modeling and exchange of product classification systems using XML. *Proceedings of the 4th IEEE International Workshop on Advanced Issues of E-Commerce and Web-based Information Systems.*

Marron, P. J., Lausen, G. & Weber, M. (2003): Catalog integration made easy. *Proceedings of the ICDE2003*, Bangalore.

Muldoon, K. (2000). *How to profit through catalog marketing*. NTC Business Books.

Olson, G. (2000). An overview of B2B integration. *Enterprise Application Integration Journal, 4*, 28-36.

Omelayenko, B. & Fensel, D. (2001). An analysis of B2B catalogue integration problems. *Proceedings of the International Conference on Enterprise Information Systems (ICEIS-2001)*, Setúbal, Portugal, July 7-10.

Porter, M. F. (1980): An algorithm for suffix stripping. *Program*, 14(*3*).

Poulovassilis, A. & McBrien, P. (1998). A general formal framework for schema transformation. *Data & Knowledge Engineering, 28,* 47-71

Quantz, J. & Wichmann, T. (2003). E-business-standards in Germany - Research project commissioned by the German Federal Ministry of Economics. Final report (Short version). Available at http://www.berlecon.de/studien/InhaltProbe/200304eStandardsKF_en.pdf

RosettaNet (2004). RosettaNet technical dictionary: RNTD Specification v4.0. 2004. Retrieved 20 January 2005from http://www.rosettanet.org

Schulten, E., Akkermans, H., Botquin, G., Dörr, M., Guarino, N., Lopes, N., & Sadeh, N.. (2001, July/August). The E-commerce product classification challenge. *IEEE Intelligent Systems Magazine.* [Special issue on Intelligent E-business].

Storeserver. (2005). *Storeserver Classifier 1.5*, *Handbook.* Available at http://www.storeserver.com

Tannenbaum, A. (1996). *Computer networks*. Pearson US Imports & PHIPEs.

Wolin, B. (2002). Automatic classification in product catalogs. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Biographies

Sven Abels works as a research assistant at the working group business information systems at the University of Oldenburg. Sven Abels is currently working on his PhD thesis at the department of computer science in Oldenburg. His main research areas are classification systems within the domain of electronic product catalogs. He is furthermore interested in interoperability- and integration-aspects within modern e-Business. Mr. Abels may be contacted at abels@wi-ol.de

Axel Hahn is head of the working group business information systems at the University of Oldenburg in northern Germany. He holds a junior professorship at the business information systems department. His main research areas are interoperability in virtual organizations and information processing at the product development. Mr. Hahn may be contacted at hahn@wi-ol.de