Integrating Vocabularies: Discovering and Representing Vocabulary Maps

Borys Omelayenko

Vrije Universiteit, Division of Mathematics and Computer Science
De Boelelaan 1081a,1081hv, Amsterdam, The Netherlands
www.cs.vu.nl/~borys
borys@cs.vu.nl

Abstract. The Semantic Web would enable new ways of doing business on the Web that require development of advanced business document integration technologies performing intelligent document transformation. The documents use different vocabularies that consist of large hierarchies of terms. Accordingly, vocabulary mapping and transformation becomes an important task in the whole business document transformation process. It includes several subtasks: map discovery, map representation, and map execution that must be seamlessly integrated into the document integration process. In this paper we discuss the process of discovering the maps between two vocabularies assuming availability of two sets of documents, each using one of the vocabularies. We take the vocabularies of product classification codes as a playground and propose a reusable map discovery technique based on Bayesian text classification approach. We show how the discovered maps can be integrated into the document transformation process.

1 Introduction

Historically business integration has been performed via costly Value-Added Networks (VANs) that use private exchange protocols and provide full range of network services for large companies. The structure of the messages and documents exchanged with VANs is specified according to the EDI X12 standard[1] that defines the structure for around 1000 plain text business documents. This architecture is a proven expensive but successful solution for large companies. However, each EDI implementation requires substantial labor effort to program and maintain, and this makes EDI solutions unacceptable for small and medium enterprises (SME) searching for cheap and easy integration solutions. SME tend to use Internet instead of costly VANs and are more flexible than the large companies in using XML-based standards for document exchange.

EDI suffers several generic document representation problems: unclear and complicated document syntax of plain text position-based formatting, unreadable semantics of document elements, weakly defined vocabularies of element

[1] www.disa.org

I. Horrocks and J. Hendler (Eds.): ISWC 2002, LNCS 2342, pp. 206–220, 2002.

values, and absence of any formal semantics of the documents. XML technologies provide a partial solution to these problems with unified syntax, implicit means for vocabulary representation, and explicit naming facilities for document elements. A number of XML-based document standards have recently been proposed trying to provide an explicit XML markup of documents and a number of tools have been developed to help in mediating between different EDI documents via XML, e.g. MS Biztalk[2]. They allow programming wrappers to translate EDI document structures to XML DTDs and then transform instance documents with XSLT [1]. However, XSLT stylesheets produced by these tools need to align different XML syntaxes, different data models, various vocabularies of XML tag names and their values, document places in a business process. An attempt to implement all these tasks with XSLT without making the semantics of document explicit leads to creation of very complicated, non-reusable and not maintainable stylesheets.

A more advanced approach [2] adopts the general idea of the Semantic Web to annotate the documents with machine-processable semantics and perform document processing based on this semantics. It assumes that first the documents are transformed from XML representation into their conceptual models in RDF [3]. Second, the models are mapped to a mediating ontology specifying shared formal semantics for each concept and containing a process ontology that specifies the order and dependency between the documents. Regular vocabularies used in the documents, e.g. product or country codes, are independently aligned to the vocabularies used in the mediating ontology. Furthermore, we need to separate between three different integration subtasks: document transformation, vocabulary mapping, and process aligning.

Vocabulary maps are large in size and homogeneous in structure, and their reuse seems to be a very efficient and relatively easy task. Automated map discovery techniques can be developed and successfully used because of the large size of vocabularies and availability of a great amount of documents using these vocabularies.

The products mentioned in product catalogues and other business documents are usually classified according to a certain product classification standard. Product codes need to be changed during the document transformation process if a pair of enterprises uses different product classification standards [4] but needs to exchange business documents. These standards form a kind of vocabularies and we use them as a playground and propose a reusable map discovery technique based on Bayesian text classification approach. In a certain sense this paper can be treated as a response to the product reclassification challenge [5] targeting a specific but very important and frequent task of product reclassification.

We define a vocabulary as a hierarchy of *terms* without multiple inheritances, constraints, properties, and other ontological primitives. Each term has an associated *term description* that specifies a free-text document associated with the term. In many cases lots of documents using a certain vocabulary are available at the companies, and each document can be treated as an extended description

[2] http://www.biztalk.org

of the term used in the document. In this paper we discuss the process of discovering the maps between two vocabularies assuming availability of two sets of documents, each using one of the vocabularies.

The paper is organized as follows: the product cataloguing task is described in Section 2 and the algorithm for map discovery is presented in Section 3. In Section 4 we present a roadmap for incorporating the maps into the document transformation process. The paper ends up with conclusions and discussion on related work in Section 5.

2 The Product Cataloguing Task

In the product cataloguing task [4] the documents represent free-text product descriptions classified according to a certain product encoding standard. The standards contain hierarchies of product categories used by the users to browse and search the collection of product descriptions.

The well-known product encoding standard UNSPSC[3] contains about 20.000 categories organized into four levels of taxonomy. Each of the UNSPSC category definitions contains only a category name with a short single-line description provided by the standardizing organization, e.g. category 43171903 ‘Central Processing units, motherboards, or daughterboards’ (a subcategory of category 431719).

The descriptions of actual products as they appear in product catalogs are also short and specific. A pair of typical product descriptions with appropriate UNSPSC codes is presented in Figure 1.

UNSPSC code	Product description
43171903	PIII 800/133 S1 256
43172402	S170B 17” 60kHz Color Monitor

Fig. 1. A typical part of a product catalogue

Another product encoding standard Eclass[4] defines more than 12.000 categories organized in a four-level taxonomy and enriched with category attributes. UNSPSC classifies the products from the supplier’s perspective while Eclass does it from the buyer’s side[5]. Another product standard NAICS[6] is used by US companies and official structures for statistical and analytical purposes, that requires specific standards used by the companies to be mapped to NAICS. The difference between the product standards can be quite substantial, e.g. more than

[3] www.unspsc.org

[4] www.eclass.de

[5] This view is not stated explicitly, however, it is unofficially supported by the standard developers and users.

[6] http://www.census.gov/epcd/www/naics.html

100 UNSPSC categories from family 43 (codes 43xxxxxx) are mapped to around ten NAICS classes. The product reclassification task assumes a supplier using one encoding standard, a buyer using another one, and a mediator maintaining the maps between both standards to perform instance data reclassification with high speed.

```
<rdfs:Class rdf:about="unspsc:43171903"
  rdfs:label="Central Processing units, motherboards, or daughterboards">
  <rdfs:subClassOf rdf:resource="unspsc:431719"/>
</rdfs:Class>
<rdfs:Class rdf:about="unspsc:431719"
  rdfs:label ="Memory and Processor Units">
  <rdfs:subClassOf rdf:resource="unspsc:4317"/>
</rdfs:Class>
```

Fig. 2. Vocabulary definition in RDF Schema

Each category in a coding standard has a standard category description, e.g. '43171903 - Central Processing units, motherboards, or daughterboards', and a place in the hierarchy. The category codes form vocabulary terms, the category definitions form primary term descriptions, and the coding standards themselves are obvious vocabularies. The sets of product descriptions correctly classified to a certain category can be treated as a secondary description of the category, and we use them to discover the maps between the categories.

It is natural to expect that RDF Schema [6], an upcoming W3C standard for representing conceptual models on the Web will be widely used to represent vocabularies on the Semantic Web. RDF Schema allows representing hierarchies of classes and properties together with possible assignments of properties to classes. Vocabulary terms can be represented with RDF Schema classes as illustrated with a fragment of UNSPSC in Figure 2. Some of the standards are already public-available in RDF Schema, e.g. UNSPSC[7]. Document conceptual models using vocabularies can be also represented in RDF Schema, as illustrated in Figure 3 and discussed in [2]. The standard category description from Figure 2 together with the product descriptions belonging to the category (e.g. 'PIII 800/133 S1 256') forms the full description of the term '43171903', as far as it is seen from our examples.

```
<itemdescription rdf:about="ITEM_00005" code="unspsc:43171903"
description="PIII 800/133 S1 256"/>
<itemdescription rdf:about="ITEM_00007" code="unspsc:43172402"
description="S170B 17'' 60kHz Color Monitor"/>
```

Fig. 3. Two product catalog items in RDF

[7] http://protege.stanford.edu/ontologies.shtml

Machine learning techniques, namely Bayesian learning has been successfully applied to assist the user in classifying new products [4] using manually pre-classified examples. In the classification setting the terms are called classes and the task of assigning the right product code given a product description is called classification of the description[8]. Machine learning techniques generate product classification rules and we use them to discover explicit mappings between vocabulary terms.

3 Discovering Vocabulary Bridges

3.1 Naive-Bayes Classifier

Recent experiences in building product classification systems [7] show that Naive-Bayes classifier can be successfully applied to classify the products and it produces sound classification rules. These rules are represented in the form of conditional probabilities defined over full category descriptions, composed of all descriptions of the products classified to the category. The descriptions consist of natural-language words, e.g. English words. The Naive-Bayes classifier (see [8], Chapter 6 for an introduction) uses two kinds of probabilities: probability $P(w_k|c_j)$ that a certain word w_k will appear in a document belonging to category c_j, and probability $P(c_j)$ of each category c_j (probability that a new product description belongs to category c_j). These probabilities are estimated from the correspondent frequencies computed over full descriptions of each category, and the result is stored in a probability table for $P(w_k|c_j)$ illustrated in Table 1. Then, for each new product description to be classified, Naive-Bayes predicts its class with the following rule:

$$prediction = argmax_{c_j} P(c_j) \prod_k P(w_k|c_j)$$

where c_j denotes a class (a product category in our case), j iterates over all classes; w_k denotes a word that appeared in the full description of the category, k iterates over all words used in the descriptions, e.g. restricted English vocabulary.

Assume that we need to map the terms between two vocabularies used in two sets of documents, and these sets are disjoint, i.e. they do not contain a single product description explicitly classified to both product classification standards. Then we assume that we are able to train the Naive-Bayes classifier on each of these two sets of documents, and we have in our possession two probability tables similar to the one depicted in Table 1. These tables will be more coherent if the document sets are overlapping and contain some products that are present in both sets.

No mapping rules can be derived from these tables if the documents use different sets of words and terms. However, this is practically a rear case: the most

[8] Different communities look at similar tasks from different perspectives and have different names for them, e.g. in the knowledge engineering area the classification task would be rather called 'identification'.

informative words in product descriptions are model names and parameters, and some mapping information may be derived even if the rest of the product descriptions is specified in different languages.

Table 1. Probability table for Naïve-Bayes (a fragment), the cells represent $P(w_k|c_j)$ in %

Classes c_j	Electric	Memory	Monitors	Notebooks
Words w_k	component	modules		
256MB	0	20	0	0
Memory	0	40	0	0
M300	0	0	0	25
S710	0	0	11	0
17	0	0	11	0
Color	0	0	44	0
Monitor	0	0	56	0

3.2 Deriving the Maps

The probability tables computed by the Naive-Bayes classifier have several peculiarities inspired by the nature of the product classification task:

- The tables tend to be very sparse with a large fraction of cells containing zeroes, and only a relatively small fraction containing non-zero values. And even in the latter case the probabilities tend to belong to a fixed set of values (e.g. 1, 0.66, 0.33). This happens because the amount of distinct words that can be used in the full category descriptions is comparable to the number of classes (about 20.000 English words are used to describe the products classified to 20.000 classes) and to the number of available product descriptions. Hence, the number of word occurrences per class can be very low and this leads to generation of very sparse tables and rough probability estimates.
- The descriptions are very short and the words used there are very specific, and very often one word indicates only one class (e.g. product model name). This is quite unusual for the text classification task where the category of a description is derived from the combination of probabilities associated to several words, where each word can, in turn, point to several classes. As a result, Naive-Bayes cannot easily find the maximal probability estimation because all the probabilities $P(w_k|c_j)$ associated to a certain class c_j can be equal.
- The small number of available descriptions per class causes high noise in the probability estimates and one or two ‘noisy’ words, which occasionally occurred in a product description, may receive the same importance as the product name itself. This somehow contradicts with the nature of Bayesian

learning that assumes the probabilities to be ‘trustable’. We need to weight the probability estimates to ensure that we will use only the probabilities calculated on the basis of sufficient number of examples. For this we weight each estimate with logarithm of the number of examples participated in computing this probability.

Let us look at the mapping discovery task from the probabilistic point of view. We denote the event that a random product description from the set D of all possible product descriptions is classified to the i-th class of the source standard with src_i and to the j-th class of the target standard with trg_j. The task of discovering a map between the source and the target classes is the task of discovering pairs of classes that maximize the probability that a random example from D will be classified to both classes, i.e. maximize the probability of co-occurrence of events src_i and trg_j: $argmax_{i,j}P(src_i \wedge trg_j|D)$.

It is easy to represent $P(src_i \wedge trg_j|D)$ via the Bayes rule and word occurrences w_k assuming that they are independent:

$$P(src_i \wedge trg_j|D) = \frac{P(D|src_i \wedge trg_j)P(src_j \wedge trg_j)}{P(D)} =$$

$$= \frac{\prod_k P(w_k|src_i \wedge trg_j)P(src_j \wedge trg_j)}{P(D)}.$$

We treat the events src_i and trg_j as independent (while, clearly, they are somehow correlated), and this allows deriving $P(w_k|src_i \wedge trg_j|D)$ via the probabilities that we have already estimated in the two Naive-Bayes classifiers trained for each of the vocabulary:

$$P(w_k|src_i \wedge trg_j|D) = P(w_k|src_i) \cdot P(w_k|trg_j)$$

We multiply the frequencies $P(w_k|c_j)$ from the probability tables by $ln(num_i)$, where num_i is the number of examples that participated in computing probability $P(w_k|c_j)$, to weight them due to the reasons mentioned above. Accordingly, we receive the final formula for discovering a one-to-one bridge between two terms from two vocabularies:

$$bridge = argmax_{i,j} \prod_k P(w_k|src_i)ln(num_i) \cdot P(w_k|trg_j)ln(num_j) \cdot P(src_i)P(trg_j)$$

where num_i and num_j denote the number of examples participated in computing probabilities $P(src_i)$ and $P(trg_j)$ respectively. $P(D)$ is omitted because it does not influence the $argmax$ result.

The amount of discovered bridges is quite high and it is difficult for the user to select the most important bridges. Accordingly, we rank each bridge t with the number of examples $num_i + num_j$ supporting the bridge:

$$Rank_t = num_i + num_j$$

where higher rank indicates a more important bridge. As a result the bridges that cover more examples get higher rank than those that cover fewer examples.

3.3 Experimental Investigation: English Dataset

For our current experiments we used two datasets of 100 examples each, the first dataset was classified according to Eclass, and the second – according to UNSPSC. Both datasets contained the products belonging to the same domain of computers and computer equipment, like the sample from Figure 1. The datasets were randomly drawn from a dataset of ten thousands products. They contained 32 UNSPSC and 20 Eclass categories that may be linked by 12 reasonable bridges, nine one-to-one and three two-to-one. 54% of the words (do not mix up with examples, i.e. product descriptions) have appeared in both document sets. The algorithm described above has derived 31 bridges, top seven of which are presented in Table 2.

Table 2. Experimental results

	UNSPSC	Eclass	Rank
1.	43171803 Desktop computers	240103 Hardware (workstation)	31
2.	43171801 Notebook computers	240109 Computer (portable)	18
3. ×	43172313 Hard drives	240103 Hardware (workstation)	13
4.	43172401 Monitors	240106 Screen	12
5. ×	43172402 Flat panel displays	240103 Hardware (workstation)	10
6.	43171802 Docking stations	240109 Computer (portable)	10
7.	43173002 Ethernet repeaters	240107 Periph. equip accessories (PC)	9

From the domain point of view most of the bridges are correct: 1,2,4,6, and 7, while 3 and 5 are wrong. These misclassifications can be easily caused by the small size of the training sets and general problems of the Naive-Bayes classifier discussed above. Interesting to mention that an attempt to remove all language-specific words (roughly one third of the words) from the datasets does not really change the results: the bridges have other order and rankings, but still the same bridges constitute the top (see Table 3).

Table 3. Experimental results: no-language dataset

	UNSPSC	Eclass	Rank
1.	43171803 Desktop computers	240103 Hardware (workstation)	31
2.	43171801 Notebook computers	240109 Computer (portable)	18
3. ×	43172401 Monitors	240103 Hardware (workstation)	10
4. ×	43172313 Hard drives	240103 Hardware (workstation)	10
5.	43171802 Docking stations	240109 Computer (portable)	10

3.4 Further Experiments

We tried the marginal case: to derive the maps between English document set and French document set (again, drawing the examples randomly, so the sets of products described in these two sets had quite a little overlap). In this experiment word overlap was equal to 18% (mostly made-up of numeric terms and model names) and as a result much less information was available to the mapping algorithm. The results are presented in Table 4. The first bridge seems to be reasonable, while the rest are not really (and also their rank is quite low). After examining the datasets we found that besides the products were belonging to the same domain (UNSPSC 4317xxxx) they were quite different in the English and French datasets.

Table 4. Experimental results: English to French bridges

	UNSPSC	Eclass	Rank
1.	43171806 Servers	240103 Hardware (workstation)	18
2. ×	43171801 Notebook computers	240107 Periph. equip accessories (PC)	14
3. ×	43171803 Desktop computers	240107 Periph. equip accessories (PC)	7

In certain high-technical domains the descriptions are very-well identifying and the choice of the language does not influence the results a lot. However, we can expect the results to be worse in other domains that use free-text descriptions of products.

Besides the recall and accuracy for the generated bridges are numerically low they are still quite significant taking into account the complexity and novelty of the problem.

4 Incorporating the Bridges in the Document Transformation Process

The vocabulary integration task forms a part of the whole business document integration process and the vocabularies and mapping rules must be represented and processed in the same way as the rest of the documents.

4.1 Representing the Maps with the Mapping Meta-ontology

Mapping knowledge represented with the maps between the categories must be represented on the Web in the machine-processable manner. Providing a certain XML serialization for the maps is not sufficient, and the serialization must be augmented with a conceptual model and formal semantics of the maps. We developed an RDFT (RDF Transformation) mapping meta-ontology to specify different mappings that occur in the business integration task. In this section we briefly sketch the main concepts of RDFT used in the vocabulary integration task and refer the reader to more extensive documents and tools available from the RDFT project homepage[9].

The main concept of the RDFT meta-ontology is the bridge between two sets of rdf:Resources (two sets of concepts, either RDF classes or properties; only RDF classes are used in the vocabulary mapping tasks), one of which is regarded as the source set, and the other one as the target set. The bridges are grouped into maps, each of which is a collection of bridges serving a single purpose. The maps are identified by their URI's and form minimal reusable modules of RDFT bridges, e.g. the maps between country names in different languages, or between product categories as discussed in this paper.

An abstract class Bridge describes common properties of the bridges and restricts them to be either one-to-many or many-to-one. Each Bridge contains a ValueCorrespondence property linking to a map aligning the values of the concepts mapped in the bridge.

The bridges can represent several possible Relations, while only the EquivalentToSet relation is used in the vocabulary integration task. EquivalentToSet bridges specify that the source set of elements is equivalent to the target set of elements, e.g. a one-to-many EquivalentToSet bridge represents the fact that the source element (single-element set) is equivalent to the target *set* of several elements, while it is not equivalent to any of the target elements alone.

Several types of Bridges are defined in RDFT:

- Class2Class and Property2Property bridges between RDF Schema classes and properties. In RDF Schema classes are represented by their names, place in taxonomy, and properties that are attached to this class. Properties are defined as first-class objects together with classes, and they capture most of domain knowledge. Classes specify aggregation of properties, and we do not include class-to-property and property-to-class bridges in RDFT believing that they will introduce a conceptual mismatch and will not provide any added value from the application point of view.
- Tag2Class and Tag2Property bridges between XML tags from the source DTD and the target RDF Schema classes and properties. Tag2Class bridges are used to link vocabulary terms represented in XML documents to the appropriate classes created to represent these terms. An example of a Class2Class bridge between product categories encoded in RDF is presented in Figure 5.
- Class2Tag and Property2Tag bridges between RDF Schema classes and properties, and the elements of the target DTD. Class2Tag bridges are necessary to link back RDF classes representing vocabulary terms to XML term representation.

The values of RDF properties and XML tags mapped with the Property2Property and Tag bridges are transformed as specified in the ValueCorrespondence maps attached to the bridges that specify an XPath [9] value transformation procedure. It can be either a DeclarativeMap or a Procedu-

[9] http://www.cs.vu.nl/~borys/RDFT

ralMap. DeclarativeMaps are ordinary maps containing Class2Class bridges, where each class corresponds to a vocabulary term.

The XMLtag class corresponds to a DTD element or attribute, identifying them with the XMLtagName and XMLtagAttributeName. RDF Schema classes and properties are used to point to the classes and properties used in the bridges. However, RDF Schema does not provide any means to represent XML tags, and we define our own class to model them.

Important to mention that cyclic maps between two terms from different vocabularies are possible and occur quite often. The cyclic map consists of a super-term Src-A, mapped to term Trg-B in another vocabulary, whose super-term Trg-A is in turn mapped to the original term Src-B. This map is represented in Figure 4 with bold curved arrows.

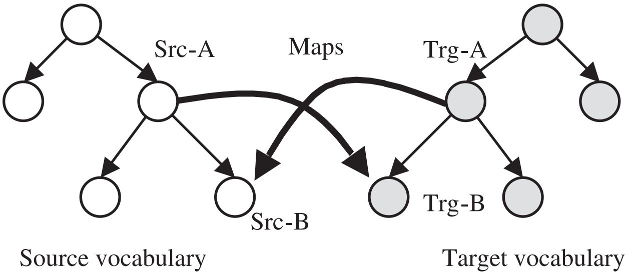

Fig. 4. Cyclic maps between the terms

In this case the map can be mistakenly interpreted as a subclass-of relationship between the terms Src-B and Trg-A, and Trg-B and Src-A. Modeling this map in such a way leads to declaring all four terms as equivalent. However, they are definitely not because two taxonomies are formed according to different classification principles. The bridges indicate that Trg-A is equivalent to Src-B and Src-A is equivalent to Trg-B but we may not derive any conclusions about Src-B and Trg-B from that.

Instance-driven semantics of the bridges leads to certain difficulties in specifying formal semantics of RDFT in terms of schema-oriented languages like DAML-OIL[10].

4.2 Vocabularies in XML Documents

Vocabularies may be encoded in XML with one of the following ways:

- An XML attribute may have a fixed list of values (so-called choice attributes) that are treated as vocabulary terms.
- An XML element may allow only EMPTY elements as its children, and the names of these tags are treated as vocabulary terms.

[10] http://www.ontoknowledge.org/oil/

```
<RDFT:Property2Property rdf:about="P2P">
        <RDFT:ValueCorrespondence rdf:resource="VC"/>
        <RDFT:SourceProperty rdf:resource="ProductCode"/>
        <RDFT:TargetProperty rdf:resource="Classification"/>
</RDFT:Property2Property>
<RDFT:DeclarativeMap rdf:about="VC">
    <RDFT:Brigdes rdf:resource="UNSPSC_ECLASS_00"/>
    <RDFT:Brigdes rdf:resource="UNSPSC_ECLASS_01"/>
</RDFT:DeclarativeMap>
<RDFT:Class2Class
rdf:about="UNSPSC_ECLASS_00">
    <RDFT:TargetClass rdf:resource="eclass:240109"/>
    <RDFT:SourceClass rdf:resource="unspsc:43171801"/>
    <RDFT:Relation rdf:resource="EquivalenceRelation"/>
</RDFT:Class2Class>
```

Fig. 5. A Property2Property bridge linking two different properties standing for the product classification code with the corresponding vocabulary map (DeclarativeMap) aligning different UNSPSC and Eclass terms via Class2Class bridges

- An XML element may have a #PCDATA type with additional knowledge that its free-text values represent the terms from a certain vocabulary (similarly, XML attributes may contain vocabulary terms as their #CDATA values).

The document integration architecture [2] envisages several transformation steps for each transformation transaction: XML-RDF transformation, several RDF-RDF steps, and the final RDF-XML transformation. Vocabulary terms can occur at two different levels: the level of document elements and the level of element values. In both cases they must be first translated to RDF representation with the Tag2Class bridges. Then the terms are aligned to the mediating vocabulary and then to the target vocabulary with the Class2Class bridges. Finally, the target XML encoding for the terms is restored with the Class2Tag bridges.

The terms remain the same while being translated from their XML serialization to RDF classes, and these steps do not require any vocabulary alignment. It is performed during the RDF-RDF transformations where different terms from different vocabularies are mapped with DeclarativeMaps.

5 Outlook and Discussion Issues

The vocabulary integration task discussed in this paper is quite specific with a number of restricting assumptions. However, this task occurs quite often in precisely the same setting that makes the solution widely reusable.

Besides the topic of vocabulary and namely catalog integration is relatively new, a certain work has been recently reported. The Naive-Bayes algorithm has been successfully applied to the task of text reclassification [10]. In this task some information about existing classification of short descriptions was used to reclassify them to another system of classes. It is similar to the task we discuss in the present paper but is not equivalent. It focuses at the issue of improving the results of product classification if the products have already been pre-classified. It assumes that the catalogs have precisely the same structure and existing classifications according to one standard can improve the classification results for another standard, while we make no assumptions on that. In addition, the approach [10] is limited to the search of one-to-one maps only. The focus was made on the artificial catalogs or news archives that have a limited number of classes and long descriptions. And the main feature of real-life product catalogs – short descriptions and huge amount of classes – is not addressed and not exploited.

In [11] different business standards are analyzed and a knowledge-engineering methodology for integrating them is sketched. From another perspective an attempt to look at the catalog integration problem from the graph-based point of view is made in [12], treating catalog structures as graphs to be aligned. We need to mention the work on automated schema integration [13], namely applied to matching XML documents [14]. Schema matching is an orthogonal task that must be solved for business integration in addition to the vocabulary mapping task.

Specifically in our work, RDFT modeling corresponds to generic OMG [15] recommendations for modeling data transformations. Conceptually an RDFT Map corresponds to TransformationMap in CWM, and the Bridge class is equivalent to the CWM's ClassifierMap (linking two concepts that are allowed to have instances). The Class2Class bridge corresponds to the source and target elements of the ClassifierMap class in CWM and our Property2Property bridge corresponds to the FeatureMap class in CWM. Due to pragmatical reasons we do not include class-to-property bridges in RDFT thus restricting CWM to a less expressive language.

We are also investigating the approaches to create tractable knowledge representation languages with limited expressiveness like CLASSIC [16] to make the mapping meta-ontology as expressive as possible without making the instance document transformation process too complex.

Let us sketch several possible future research directions in the product reclassification and vocabulary mapping area:

- The vocabularies contain the hierarchies of terms and these hierarchies must be taken into account while deriving maps between the terms. For examples, the classes assigned to product descriptions may not belong to the lowest level of the hierarchy, but to higher levels. Accordingly, a map discovery algorithm must be able to discover the maps between high-level categories, that is still not the case in our approach.
- In our approach one-to-many and many-to-one bridges are derived from one-to-one bridges by the following rule: if a pair of bridges connects several different source classes to a single target then it must be treated as a many-to-one bridge and vice versa. However, such an approach makes no difference between a many-to-one (one-to-many) bridge and an inconsistent bridge, and can be improved.

- Applicability of the Naive-Bayes classifier depends on the overlap between the words used in the documents and the overlap between product descriptions. In the marginal cases (e.g. very strong or very weak overlap) another algorithms may be needed, and an automated method selection procedure needs to be developed. We need to be able of selecting the subsets of product descriptions suitable for the bridge discovery task.

We are working now on a large-scale experimental investigation of the proposed techniques. However, it is difficult to make a good experimental setting to evaluate the results: no axiomatically correct maps are available between competing product encoding standards and their manual creation brings a certain degree of objectivism into the evaluation. However, the fact that the approach discovers valuable maps gives a hope for its stable behavior and usability. Practical usability remains a dominating quality criterion for the evaluation of our work as well as many other Semantic Web activities.

Acknowledgements

We would like to thank Dieter Fensel, Michel Klein, Maxym Korotky, Ellen Schulten, and Volodymyr Zykov for their helpful discussions and comments, and two anonymous reviewers for their comments.

References

1. Clark, J.: XSL Transformations (XSL-T). Technical report, W3C Recommendation, November 16 (1999)
2. Omelayenko, B., Fensel, D.: A Two-Layered Integration Approach for Product Information in B2B E-commerce. In Madria, K., Pernul, G., eds.: Proceedings of the Second International Conference on Electronic Commerce and Web Technologies (EC WEB-2001). Number 2115 in LNCS, Munich, Germany, September 4-6, Springer-Verlag (2001) 226–239
3. Lassila, O., Swick, R.: Resource Description Framework (RDF) Model and Syntax Specification. Technical report, W3C Recommendation, February 22 (1999)
4. Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M., Flett, A.: Product Data Integration for B2B E-Commerce. IEEE Intelligent Systems **16** (2001) 54–59
5. Schulten, E., Akkermans, H., Botquin, G., Dorr, M., Guarino, N., Lopes, N., Sadeh, N.: The E-Commerce Product Classification Challenge. IEEE Intelligent Systems **16** (2001) 86–88
6. Brickley, D., Guha, R.: Resource Description Framework (RDF) Schema Specification 1.0. Technical report, W3C Candidate Recommendation, March 27 (2000)
7. Ding, Y., Korotkiy, M., Omelayenko, B., Kartseva, V., Zykov, V., Klein, M., Schulten, E., Fensel, D.: Goldenbullet in a nutschell. In: Proceedings of the 15-th International FLAIRS Conference, Pensacola, Florida, May 16-18, AAAI Press (2002)
8. Mitchell, T.: Machine Learning. McGraw Hill (1997)
9. Clark, J., DeRose, S.: XML Path Language (XPath), version 1.0. Technical report, W3C Recommendation, November 16 (1999)

10. Agrawal, R., Srikant, R.: On Integrating Catalogs. In: The 10-th International World Wide Web Conference, Hong Kong, May (2001)
11. Corcho, O., Gomez-Perez, A.: Solving Integration Problems of E-commerce Standards and Initiatives through Ontological Mappings. In: Proceedings of the Workshop on E-Business and Intelligent Web at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, USA, August 5 (2001)
12. Navathe, S., Thomas, H., Amitpong, M.S., Datta, A.: A Model to Support E-Catalog Integration. In: Proceedings of the Ninth IFIP 2.6 Working Conference on Database Semantics, Hong-Kong, April 25-28 (2001) 247–261
13. Rahm, E., Bernstein, P.: A Survey of Approaches to Automatic Schema Matching. The VLDB Journal **10** (2001) 334–350
14. Anhai, D., Domingos, P., Halevy, A.: Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. In: Proceedings of the ACM SIGMOD Conference, Santa Barbara, CA, May 21-24, ACM (2001)
15. CWM: Common Warehouse Model Specification. Technical report, Object Management Group (2001)
16. Borgida, A., Brachman, R., McGuinness, D., Resnik, L.: CLASSIC: A Structural Data Model for Objects. In: Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data, Portland, OR, May 31 - June 2, ACM (1989) 59–67