An Analysis of Integration Problems of XML-Based Catalogs for B2B Electronic Commerce

Borys Omelayenko, Dieter Fensel
Division of Mathematics and Computer Science, Vrije Universiteit, De Boelelaan 1081a, 1081HV, Amsterdam, The Netherlands
Email: {borys, dieter}@cs.vu.nl, URL: www.cs.vu.nl/~{borys, dieter}

Abstract: Electronic business-to-business marketplaces mediate between many online suppliers and buyers, each of which can potentially use his own format to represent the products in his product catalog. The marketplaces have to perform non-trivial translations between these formats. In this paper we analyze the problems which occur during the integration, taking several leading XML and non-XML formats as examples. We discuss how the XSL-T technology can be applied to perform such type of transformation, provide the corresponding examples highlighting the problems, and propose typical solutions to these problems.

Key words: B2B Catalog Integration, Product Catalogs, XML Integration, XSL-T

1.   INTRODUCTION

Electronic marketplaces for Business-to-Business (B2B) electronic commerce bring together many online suppliers and buyers. In order to function, they require the integration of many product catalogs provided by the marketplace participants. Each individual participant can potentially use his own format to represent the products in his product catalog. If a marketplace mediates between n suppliers and m buyers, then it must be able to map each of the n suppliers' catalogs into m buyers' formats performing $n{\times}m$ mappings. The numbers n and m may be high enough to make the problem of creating and maintaining these catalog integration rules non-trivial (Fensel et al. 2001).

Different suppliers tend to use completely different representation schemas even for such a simple concept as the address. Fortunately, most of them use XML syntax in their encoding, but the structures of the descriptions differ greatly and a number of variations have already been developed and implemented. Large buyers also create and maintain their buyer catalogs and use the same set of catalog standards as the suppliers. A B2B mediator has to integrate both suppliers' and buyers' formats to allow them to do contracting with one another. This makes the problem of standard integration and interoperation a very important one (see (Li 2000) and (Ng, Yan, and Lim 2000) for an overview).

Existing logic-based schema integration approaches like TRIPLE (Sintek and Decker 2001) provide a complete solution of the transformation problem, which, however, does not scale due to high computational complexity of the logical inference process. Knowledge transformation approaches (Chalupsky 2000) solve similar problems in knowledge-based transformation, but they focus at different technologies and application scenarios and do not employ the full power of XML-related technologies. W3C Consortium[1] provides an XML low-level integration architecture with XSL-T (Clark 1999) and XPath (Clark and DeRose 1999) languages. Given the importance and good perspectives of XML, a successful catalog transformation technology must be built on top of it. This justifies the focus of further development of XML transformation technology.

Leading enterprise document transformation tools (e.g. Biztalk[2]) show the current industrial focus in this area. They primary address the problem of integrating the documents with different plain text syntaxes, and a unified XML syntax is used as a mediating representation. All the transformations between different standards are done at the XML level with plain XSL-T technology. Practical applications of XSL-T lead to improvement efforts, e.g. by imposing additional modularity of the rules proposed in (Eder and Strametz 2001) to improve their reusability.

In this paper we discuss the methods for applying XSL-T technology to B2B document interchange, with the guidance of the previous research results mentioned above. We take the integration of address descriptions as a running example. Section 2 describes four different ways to encode an address in XML, and Section 3 discusses the model for catalog integration. Section 4 discusses several possible types of integration rules and gives examples for each type. The paper ends with some observations on the integration of non-XML catalogs in Section 5, before arriving at its conclusions.

---

[1] www.w3c.org
[2] www.biztalk.org

2. FOUR EXAMPLES OF XML CATALOGS

We take the problem of *address* integration as a running example throughout the paper. An address is a simple business concept that occurs very often in e-commerce, and it is an important part of any B2B mediation system. Unlike most of the products, the structure of an address and the meaning of its components are understandable to everybody and this makes the explanation clear. The integration of address descriptions also involves several interesting types of problems that also occur in product integration.

These problems were studied in (Li 2000) as a basic for comparison of seven product description standards and a discussion about general catalog interoperation issues. We take four address description standards as our running example and use them to continue the investigation of the integration problems.

The first standard analyzed here is *xCBL 3.0* developed by Commerce One[3], Inc. It provides a comprehensive set of standardized XML document formats, allowing buyers, suppliers, and service providers to integrate their existing systems quickly and efficiently in the electronic marketplaces (Commerce One 2000). The Document Type Definition (DTD) for an address in the xCBL standard is presented in *Figure 1*.

```
<!ELEMENT OrganizationAddress ((AddressType)?, (ExternalAddressID), (POBox)?,
(Street)?, (HouseNumber)?, (StreetSupplement1)?, (StreetSupplement2)?, (PostalCode)?,
(City), (Country), (Region)?, (District)?, (County)?, (TradingPartnerTimezone)?)>
<!ELEMENT AddressType ((AddressTypeCoded), (AddressTypeCodedOther)?)>
```

Figure 1. xCBL DTD sample

The second standard, the *Internet Open Trading Protocol (IOTP)* was developed within the Internet Engineering Task Force (IETF[4]) consortium, and it provides a standard framework for payment operations for Internet commerce. It is independent of any specific payment system. IOTP provides the data structures and communication protocols for payment transactions: purchase, refund, authentication, deposit, and other protocols that occur in electronic commerce (Burdett 1999). Security, authentication, and digital signatures are its main concerns, while the address description is not as elaborate as in the previous example. The DTD for an address in IOTP standard is presented in *Figure 2*.

---

[3] http://www.commerceone.com/
[4] http://www.ietf.org

```
<!ELEMENT PostalAddress EMPTY>
<!ATTLIST PostalAddress
    xml:lang NMTOKEN #IMPLIED
    AddressLine1 CDATA #IMPLIED
    AddressLine2 CDATA #IMPLIED
    CityOrTown CDATA #IMPLIED
    StateOrRegion CDATA #IMPLIED
    PostalCode CDATA #IMPLIED
    Country CDATA #IMPLIED
    LegalLocation (True | False) "False">
```

Figure 2. IOTP DTD sample

The Open Applications Group is an industrial consortium publishing specifications for business content in the enterprise applications space, with the emphasis on *Open Applications Group Integration Specification (OAGIS)*[5], specifications for business messages and integration scenarios. It provides data structures, messaging formats and protocols for business integration (OAGIS 2000). OAGIS defines a vocabulary of business terms and more than 90 different types of business documents can be exchanged. The OAGIS standard defines the address as presented in *Figure 3*.

```
<!ELEMENT ADDRESS (ADDRLINE*, ADDRTYPE?, CITY?, COUNTRY?, COUNTY?,
DESCRIPTN?, FAX*, POSTALCODE?, REGION?, STATEPROVN?, TAXJRSDCTN?,
TELEPHONE*, URL?, USERAREA?)>
```

Figure 3. OAGIS DTD sample

Version 1.0 of the Real Estate Data Interchange Standard (RETS 2000) defines the interchange of real estate information. It defines a standard interface by which a client program may communicate with a property or other real estate data server. The specification defines a protocol for implementing transactions, and incorporates an XML specification for general-purpose interchange. It also provides a compressed data interchange format and specification to allow the interchange of machine-interpretable property information. The data structures for the interchange are defined in the *Real Estate Transaction Markup Language (RETML)*[6], where the address is as presented in *Figure 4*.

```
<!ELEMENT MailingAddress (StreetAddress)>
<!ELEMENT StreetAddress ((StreetNumber?, BoxNumber?, StreetDirPrefix?, StreetName,
StreetAdditionalInfo?, StreetDirSuffix?, StreetSuffix?, UnitNumber?, City?,
StateOrProvince?, Country?, PostalCode?, CarrierRoute?) | Unstructured)>
```

Figure 4. RETML DTD sample

The representations of the same concept, the address, differ in each catalog. Product description can be encoded in XML with different ways of using XML tags, i.e. product features can be represented with XML attributes (like the IOTP catalog shown in *Figure 5*), or with XML elements (like the OAGIS catalog in *Figure 6*). Conceptually equal product properties can be encoded with XML elements with different names. The elements marked up with the same XML tags can have different semantics. The order of tags is also important in XML. Finally, some product properties can be described with different granularity level as required by the application. For example the focus of a delivery system requires partitioning of a company address into street name, house number, city district, an so on, while for an electronic payment system these details are not so important, as can be seen in the general-purpose xCBL and OAGIS systems, and in the IOTP payment system. At the same time real estate agencies require more details in the street description, as it appears in the RETML standard.

```
<PostalAddress AddressLine1="Division of Mathematics and Computer Science"
AddressLine2="De Boelelaan 1081a" CityOrTown="Amsterdam" Country="Netherlands"
LegalLocation="True" PostalCode="1081 hv" StateOrRegion="Horth Holland"
xml:lang="en"/>
```

Figure 5. IOTP catalog example

```
<ADDRESS>
   <ADDRLINE>De Boelelaan, 1081a</ADDRLINE>
   <ADDRTYPE>office</ADDRTYPE>
   <CITY>Amsterdam</CITY>
   <COUNTRY>Netherlands</COUNTRY>
   <COUNTY>North Holland</COUNTY>
   <DESCRIPTN/>
   <FAX>3120444765</FAX>
   <POSTALCODE>1081 hv</POSTALCODE>
   <REGION/>
   <STATEPROVN/>
   <TAXJRSDCTN/>
   <TELEPHONE>3120444765</TELEPHONE>
   <URL>www.cs.vu.nl</URL>
   <USERAREA/>
</ADDRESS>
```

Figure 6. OAGIS catalog example

3. THE MODEL FOR CATALOG INTEGRATION

If the marketplace mediates between n suppliers and m buyers, then it must be able to map each of the n suppliers' catalogs into m buyers' formats performing $n \times m$ mappings (*Figure 7*). The standard solution for such a problem is in introduction of a mediator, which provides a unified terminology. Accordingly, we introduce a mediating catalog, which we call the *Unified Catalog* (the UC). As a result, each supplier or buyer needs to maintain only one link, between its catalog and the UC, and therefore requires only $n+m$ mappings (*Figure 8*).

The UC must provide sufficient number of catalog attributes for a marketplace to perform meaningful document transformation. From another side, the UC must provide the minimal amount of attributes not to overload the users with useless transformations.
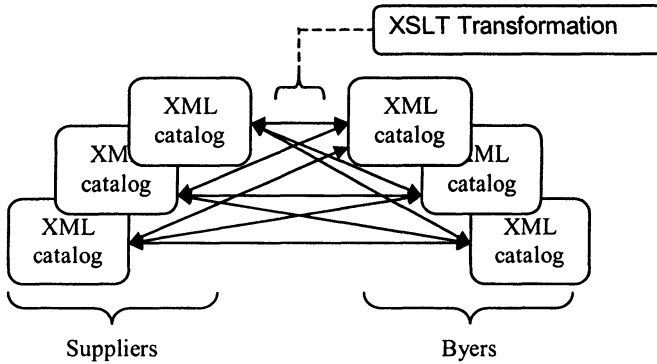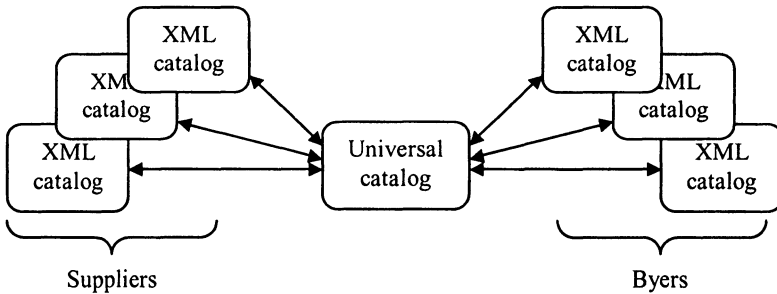

Figure 7. Integration with $n \times m$ rules

Figure 8. Integration with $n+m$ rules

In our address example, the IOTP standard partitions street information into AddressLine1 and AddressLine2 fields, while other catalogs partition it as a Street name and a House number as presented in the UC, as we will show later. In this case the partitioning of information between AddressLine1 and AddressLine2 is not defined, and AddressLine1 is not required to be equal to Street and AddressLine2 to House. A user of the IOTP standard can freely partition his street and house information between these address lines. Weak defined semantics is the reason to not include redundant elements into the UC.

The modeling primitives of the catalog systems are compared in *Table 1* together with their synonyms as they appear in the UC. We have omitted some implementation-specific attributes that do not deliver any content, i.e. system-specific identification numbers for the records. The columns in the table refer to different catalog systems, each row representing a group of synonymous primitives. The first row below the header refers to the root elements of the addresses that are presented in bold. For example, the table shows that street name is called Street in xCBL, StreetName in RETML, and Street in the UC. ADDRLINE in OAGIS corresponds to the pair (AddressLine1, AddressLine2) in IOTP and to the pair (Street, House) in the UC. Non-obligatory elements of the catalogs are marked with '?'.

In a nutshell, mapping between different standards has the following main features:

1. The catalogs contain a kernel of well-mapped elements that are present in all catalogs and represent the most important features of the entity described (i.e. Street, House, PostalCode, City, Region, and Province in our address example).
2. The catalogs contain a number of mappings between rarely used elements that represent the features which are important for one agent but not for others and which may be included in the descriptions. To keep the UC at the most detailed level, we must map these elements and this creates a sparsely populated part of the mapping table, like the Fax, Phone, etc. elements in our example.
3. The catalogs contain a jumble of ill-defined and badly shaped concepts, which are grouped and mapped in one concept of the UC, like the Description element in our example. We refer to these tags with weak real-world semantics as ill-defined concepts.

Table 1. Element Mapping

CBL	IOTP	OAGIS	RETML	UC
OrganizationAddress	PostalAddress	ADDRESS	StreetAddress	address
Street?			StreetName	Street
HouseNumber?			StreetNumber?	House
	AddressLine1	ADDRLINE		

CBL	IOTP	OAGIS	RETML	UC
	AddressLine2			
POBox?	-	-	BoxNumber?	PObox
PostalCode?	PostalCode	POSTALCODE ?	PostalCode?	Postcode
City	CityOrTown	CITY?	City?	City
Region?	StateOrRegion	STATEPROVN?	StateOrProvince?	Province
District?	-	REGION?	-	Region
County?	-	COUNTY?	-	District
Country	Country	COUNTRY?	Country?	Country
StreetSupplement1?	LegalLocation	DESCRIPTN?	StreetDirPrefix?	Description
StreetSupplement2?		ADDRTYPE?	StreetAdditionalInfo?	
			StreetDirSuffix?	
			StreetSuffix?	
			UnitNumber?	
-	-	FAX?	-	Fax
-	-	TELEPHONE?	-	Phone
-	-	URL ?	-	URL
-	-	-	CarrierRoute ?	Route
-	xml:lang	-	-	xml:lang
-	-	TAXJRSDCTN?	-	TaxDistrict
-	-	USERAREA?	-	Area
TradingPartner Timezone?	-	-	-	TimeZone

When creating the UC we must strike a balance between our desire to make it as precise as possible and avoiding the creation of lots of redundant mappings for synonymous concepts.

4. INTEGRATION AT THE XML CATALOG LEVEL

Mapping rules translate the descriptions between two catalog formats, one of which is the UC. We assume that both catalogs are represented in XML, and we use the XSL-T language to encode them. In the remaining part of this section we discuss possible cases of mapping rules between two catalogs C1 and C2 (where in practice one of them is the UC).

Four types of mapping between the attributes of C1 and C2 are possible: one-to-one mapping (1:1), one-to-many (1:n), many-to-one (n:1), and many-to-many (n:n) mapping. One-to-one mapping is known in the field of database integration as a mapping equivalence, while the last three types of mappings can be viewed as examples of transformational equivalence.

One-to-one mapping is the simplest and most common type of mapping between the elements of C1 and C2. It occurs when the element of C1 has a semantic equivalent in C2, i.e. element Region in the xCBL standard is equivalent to StateOrRegion in IOTP, to REGION in OAGIS, to StateOrProvince in RETML, and to Province in the UC. Translation rules in this case are quite simple. If the element is encoded by an XML element in both C1 and C2, then the rule can be expressed as follows (from RETML to UC):

```
<xsl:for-each select="StreetAddress">
  ...
  <Province><xsl:value-of select="StateOrProvince"/></Province>
  ...
</xsl:for-each>
```

If the element is encoded by an XML attribute in C1 and by an XML element in C2 then the rule can be expressed as follows (from IOTP to UC):

```
<xsl:for-each select="PostalAddress">
  ...
  <Province><xsl:value-of select="@StateOrRegion"/></Province>
  ...
</xsl:for-each>
```

Finally, we can translate the element encoded as an XML element in C1 (as is the case for the UC) into an XML attribute with the following rule (from UC to IOTP):

```
<xsl:for-each select="address">
  <xsl:element name="PostalAddress">
  <xsl:attribute name="StateOrRegion"><xsl:value-of select="Province"/></xsl:attribute>
  ...
  </xsl:element>
</xsl:for-each>
```

One-to-many mapping occurs when an element in C1 has to be translated into several elements in C2. For example, ADDRLINE in OAGIS semantically corresponds to the pair of attributes Street and House in the UC. XSL-T language provides the means to represent mapping on the level of XML elements and attributes, as well as possibilities of analyzing text inside an element in order to split the element into two or more pieces. Both processes are carried out by means of the XPath language (Clark and DeRose 1999). Accordingly, XSL-T rules must be extended with small XPath expressions (element parsers) that will split the elements as required. For example, in the following fragment of an OAGIS address it is assumed that ADDRLINE contains street name separated from the following house number by a comma:

```
<ADDRLINE>De Boelelaan, 1081a</ADDRLINE>
```

First ADDRLINE is split into a pair of XML elements:

```
<ADDRESS>
  <ADDRLINE>
      <STREET>De Boelelaan</STREET>
      <HOUSE>1081a</HOUSE>
  </ADDRLINE>
  ...
</ADDRESS>
```

This can be done using the following XSL-T rule:

```
<STREET>
  <xsl:variable name="addrline" select="ADDRLINE"/>
  <xsl:value-of select="substring-before($addrline,',')"/>
</ STREET >
<HOUSE>
  <xsl:variable name="addrline" select="ADDRLINE"/>
  <xsl:value-of select="substring-after($addrline,', ')"/>
</ HOUSE >
```

The new pair of tags is then translated into the UC with a pair of one-to-one rules for STREET and HOUSE separately. It is evident that many more cases of encoding are possible and require separate parsing rules.

Many-to-one mapping occurs when two or more elements from C1 have to be translated into one element in C2. For example, the Street and House elements in the UC must be translated into the element ADDRLINE in OAGIS. This can be done by means of XSL-T in the following way:

```
<xsl:for-each select="address">
  <ADDRLINE><xsl:value-of select="Street"/>, <xsl:value-of select="House"/>
  </ADDRLINE>
  ...
</xsl:for-each>
```

which will map a pair (Street, House) of UC elements into the following OAGIS record:

```
      <ADDRLINE>De Boelelaan, 1081a</ADDRLINE>
```

Many-to-many mapping occurs when a piece of a description is spread over several elements without evident partitioning of information between them. For example, Street, House, and PObox elements of the UC (which directly maps into xCBL and RETML) correspond to the pair (AddressLine1, AddressLine2) in IOTP without any indication where street, house, and postbox information should be stored within these two address lines.

Mapping of a structured UC record into a less structured IOTP record can be done straightforwardly:

```
<xsl:for-each select="address">
    <AddressLine1><xsl:value-of select="Street"/>
    <xsl:value-of select="House"/></AddressLine1>
    <AddressLine2>P.O. Box <xsl:value-of select="PObox"/> </AddressLine2>
    ...
</xsl:for-each>
```

Mapping back from an IOTP record into a structured UC record requires two steps: (1) processing of the AddressLine1 and AddressLine2 IOTP elements with a small XPath parser that creates the Street, House, and PObox elements out of them, similar to one-to-many mapping; (2) mapping of the latter three elements using one-to-one rules.

We have analyzed *Table 1* to estimate the ratio between the mappings of each type. For each catalog standard we have calculated the number of mappings of each type required to map the address from the catalog into the UC. The results are presented in *Table 2*. The ratio of the reverse mappings from the UC into the individual catalog reflects the partitioning of the straight mappings listed in *Table 2*: if an element was mapped into the UC with one 1:n mapping then the reverse mapping will require one n:1 mapping. With this in mind, we have not included the statistics of reverse mappings in the table. From *Table 2* we can see that most of the rules (89%) represent one-to-one mappings, while the other types only appear in special cases, once or twice for each catalog standard.

Table 2. The ratio of mappings by type

	xCBL to UC	IOTP to UC	OAGIS to UC	RETML to UC	Ratio
1:1	11	7	12	9	89%
1:n	0	0	1	0	2%
n:1	1	0	1	1	7%
n:n	0	1	0	0	2%

5. INTEGRATION OF NON-XML CATALOGS

There are several non-XML catalog formats already approved and used by the industry. The first is the well-known EDIFACT format, approved by the United Nations Economic Commission for Europe[7] (ASC 2000). An EDIFACT document presented as a complicated formatted text is incomprehensible to a non-specialist ('EDIFACT is often called "the nightmare of the paperless office" when you show a programmer the

[7] http://www.unece.org/trade/untdid/welcome.htm

standard 2,700-page draft'[8]). The first few lines of a purchase order in EDIFACT look like this:

```
UNA:+.? 'UNB+UNOC:2+STUB+BLA+960209:0843+72'UNH+000
0090001+ORDERS:D:93A:UN:EAN007'BGM+220+B00404'DTM+1
37:19960209:102'NAD+BY+++STADT- UND UNIVERSITAETSBI
BLIOTHEK:FRANKFURT+BOCKENHEIMER LANDSTR. 134-138+FR
...
```

Several attempts at translating an EDIFACT catalogue into XML are now available. The XML-EDIFACT[9] initiative developed an XML syntax for EDIFACT and a freely-available wrapper programmed in Perl which transfers EDIFACT documents to their XML representation, and vice versa. UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business)[10] has announced that it is developing a standard XML format for EDIFACT[11] but no results are yet available.

Another standard is ISO 10303-41 (ISO 2000) (also known as a part of STEP family of standards), which is an International Standard for the computer-interoperable representation and exchange of product data. It provides a neutral mechanism capable of describing product data throughout the lifecycle of a product, independently of any particular system. It provides a set of schemas for product data: an application context schema, a product definition schema, a product property definition schema, a product property representation schema, and a number of supporting schemas. For example an address is defined in STEP as follows:

```
ENTITY persons_in_organization_address
SUBTYPE OF (personal_address, organizational_address)
WHERE
WR1: SIZEOF(SELF\organiz_address.organizations)=1;
END_ENTITY;
```

The standard contains a rich set of modeling primitives which allows the building of hierarchical product specifications (product schemas) that generally correspond to those of XML Schema language (Fallside 2000). ISO has developed an XML syntax for the STEP standard, which is now being standardized as Part 28 of the ISO 10303 specification (Shaw 1999).

XML representations have already been developed for both standards and are now in the process of being standardized and obtaining the approval of the community. We do not expect the problem of non-XML product standards integration to be a major issue in the near future.

---

[8] http://www.xml-edifact.org/pub/
[9] http://www.xml-edifact.org/
[10] http://www.unece.org/cefact/
[11] http://www.unece.org/cefact/xml/press/eforms.pdf

6. CONCLUSIONS

Integration of product catalogs entails four types of mapping rules: one-to-one, many-to-one, one-to-many, and many-to-many mappings. One-to-one mappings occur most often (89% in our case study), and together with many-to-one rules can be straightforwardly encoded using XSL-T rules. One-to-many and many-to-many rules require special processing of the elements. This can be done using XPath language together with XSL-T, several examples of which have been given.

This paper discussed problems and solutions to map various standards in order to make B2B electronic commerce workable. The need for these mappings arises at several levels[12]:

- Different standards for describing products (e.g. UNSPSC[13], ecl@ss[14], and UCEC[15]).
- Different standards for describing product catalogues and exchangeable business documents (e.g. XML Common Business Library xCBL[16] versus Commerce XML cXML[17]).
- Different business process standards (e.g. mapping private processes to ebXML[18]).

In consequence, framework and tool environments are needed that allow the effective and efficient definition of such mappings. This framework must provide:

- A simple language on top of XSL-T customized to the specific needs for mapping rules in electronic commerce. Instead of defining complex XSL-T transformations by hand, they should be derivable from the definition of mappings on a more intuitive level.
- A distinction between syntactic and semantic transformations and a conceptual description to define the mappings. Normalizing various XML styles into a common data model is one step in the mapping process (e.g. from attributes to subentities and vice versa). A second step is the actual mapping of the semantics of the information. We are currently developing a layered framework that distinguishes different aspects in the mapping process, allowing us to identify simple and reusable mapping patterns.

---

[12] Not to mention the problem of mapping XML-based standards for updating product catalogues.
[13] http://www.unspsc.org
[14] http://www.eclass.de
[15] http://www.ucec.org
[16] http://www.xcbl.org
[17] http://www.cxml.org
[18] http://www.ebxml.org

– Finally, a unifying product catalogue requires a richer representation format as it has been discussed in this paper. Instead of simply defining it as a flat XML structure we intend to employ representation techniques from the area of *Ontologies* (Fensel 2001) to structure product and catalogue descriptions via class definitions, is-a hierarchies, and attributes accompanied by domain and range restrictions. This simplifies the integration process of heterogeneous descriptions and will enable us to make use of advanced representation and mapping tools developed in the area of ontology development (see for example (Fensel et al. 2000), (Noy and Musen 2000) and (Chalupsky 2000)).

Concisely, B2B marketplaces and their content managers help suppliers and buyers to overcome their exponential mapping problem. However, the market participants will only be able to handle this problem effectively if they can develop appropriate technology that allows them to define such mappings easily. We are currently developing a mapping approach in a close cooperation with Content Europe, which is one of the leading solution providers for Content management in B2B electronic commerce.

ACKNOWLEDGEMENTS

REFERENCES

ASC X12 Standard, version 4 release 4; *Data Interchange Standards Association, Inc.*, 2000.

Burdett, D., Internet Open Trading Protocol – IOTP, Version 1.0, In *Proceedings of the 46th Internet Engineering Task Force*; Hosted by Nortel Networks, Inc., Washington, D.C., USA, 7-12 November 1999; available online at http://www.ietf.org/proceedings/99nov/index.html.

Chalupsky, H., OntoMorph: A Translation System for Symbolic Knowledge. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 17th International Conference (KR-2000)*, Breckenridge, Colorado, USA, April 12-15, 2000, pp. 471-482.

Clark, J., DeRose, S., XML Path Language (XPath), Version 1.0, *W3C Recommendation*, 16 November 1999; available online at: http://www.w3.org/TR/xpath

Clark, J.: XSL Transformations (XSL-T), *W3C Recommendation*, November 1999; available online at http://www.w3.org/TR/xslt/.

Commerce One, XML Common Business Library (xCBL) 3.0, *Press release made by Commerce One, Inc. at the eLink Conference*, Hong Kong, 29 November 2000; available online at http://www.commerceone.com/news/us/xcbl30.html

Eder, J. and Strametz, W., Composition of XML-Transformations. In *Proceedings of the Second International Conference on Electronic Commerce and Web Technologies (EC-WEB 2001)*; LNCS, 2115:71-80, Munich, Germany, September 4-6, 2001.

Fallside, D., *XML Schema Part 0: Primer*, W3C Candidate Recommendation 24 October 2000; available online at: http://www.w3.org/TR/xmlschema-0/

Fensel, D., Horrocks I., van Harmelen F., Decker S., Erdmann M., Klein M., OIL in a Nutshell, In *Proceedings of the 12th International Conference Knowledge Engineering and Knowledge Management (EKAW 2000)*; LNAI, 1937:1-16, Springer-Verlag, 2000.

Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M., and Flett, A., Product Data Integration for B2B E-Commerce. *IEEE Intelligent Systems*; 16(4):54-59, 2001.

Fensel, D., Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag, Berlin, 2001.

ISO: Integrated generic resource: Fundamentals of product description and support; International Standard ISO 10303-41, Second Edition, 2000.

Li H., XML and Industrial Standards for Electronic Commerce. *Knowledge and Information Systems*, 2(4): 487-497, 2000.

Ng, W., Yan, G., Lim, E., Heterogeneous Product Description in Electronic Commerce. *SIGecom Exchanges, Newsletter of the ACM SIG on E-commerce*; 1, 2000.

Noy, N., Musen, M., PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*; Austin, Texas, July 30 - August 3, 2000.

Open Applications Group Inc., *Open Applications Group Integration Specification*, OAGIS Release 7.0.2, 2000; available online at: http://www.openapplications.org/

The Real Estate Transaction Standard (RETS), 2000; available online at: http://www.rets-wg.org/docs/

Shaw, N. (ed.), *Product data representation and exchange: Implementation methods: XML representation of EXPRESS-driven data*; Working Draft ISO/WD 10303-28 from 21 December 1999; available online at: http://www.pdtsolutions.co.uk/standard/xml-express/

Sintek, M. and Decker, S., TRIPLE - An RDF Query, Inference, and Transformation Language. In *Proceedings of the Workshop on Deductive Databases and Knowledge Management (DDLP-2001)*; Tokio, Japan, October 20-22, 2001.