In: Proceedings of the Second International Conference on Electronic Commerce and Web Technologies (EC WEB-2001), Munich, Germany, September 4-6, 2001, © Springer-Verlag (to appear)

A Two-Layered Integration Approach for Product Information in B2B E-commerce

Borys Omelayenko and Dieter Fensel

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

Abstract. Electronic B2B marketplaces bring together 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 mappings of these catalogs. In this paper, we analyze the problems which occur during the integration, taking several leading XML-based standards as an example. We advocate a three-layer product integration framework to resolve the difficulties in overcoming these problems with a direct one-layer integration. In this paper, we focus on the first two layers: the XML-based *syntax layer* and the *data models layer* expressed in RDF. The approach operates in three main steps. First, we create an RDF data model from the XML catalog, which eliminates all syntactical peculiarities of the catalog. Second, the catalog is translated from the source model to the RDF model of the target catalog. Finally, the transformation from RDF to XML restores all syntactical regulations required by the target catalog format. The approach is suitable for inter-operation with higher-level document and workflow ontologies.

1 Introduction

The World Wide Web has drastically changed the on-line availability of information and the amount of information exchanged electronically. The web has revolutionized personal information access and knowledge management in large organizations (cf. [7]). In addition, it has begun to change the commercial relationships between suppliers and customers. The [17] estimates for the business-to-consumer (B2C) area range between $4 billion to $14 billion on-line sales in the US for 2000, which is approximately 1% of the overall sales figures. This is still a small fraction of the overall business figures, but we can expect its fast growth given the fact that the number of Internet users grew from 100 to 300 million between 1997 and 2000. Similar estimates have been made for the business-to-business (B2B) area. Forecasts for the dollar value of B2B EC in the US range between $600 billion to $2.8 trillion for 2003 (cf. [17]). Currently, a large fraction of B2B transactions is still realized by traditional non-Internet networks, such as those conducted over EDI systems. In this traditional paradigm, direct one-to-one connections and mappings are programmed based on standards, such as EDIFACT[1]. However, this traditional paradigm does not employ the full power of electronic commerce and it will soon be replaced by the Internet and Web-based transaction types.

Electronic marketplaces for Business-to-Business (B2B) electronic commerce bring together many online suppliers and buyers, which participate in business interactions (cf. [6] for an overview of the field). Internet and web-based electronic commerce provide a much higher level of *flexibility* and *openness*, which helps to optimize business relationships. According to the U.S. Department of Commerce [17] estimates, there were around 800 B2B marketplaces in early 2000. Other studies estimate around 10,000 B2B marketplaces in the very near future. These marketplaces provide completely new opportunities for their clients:

- A supplier, linked to a marketplace is automatically linked to a large number of potential customers, instead of implementing one-to-one links to each supplier.
- A supplier or a customer can choose between a large number of potential business partners and can optimize his business relationships.

Concisely, B2B marketplaces are a middleware, which helps the customers to contact a large number of potential clients without running into the problem of implementing a large number of communication channels. However, preventing the customers from the bottleneck of the exponential growth in the number of implemented business connections becomes a serious problem for B2B marketplaces. They contend with the problem of heterogeneity in the *product*, *catalogue*, and *document* description standards of their customers. Efficient management of different description styles becomes a key task for these marketplaces. In addition, a number of serious mapping problems need to be solved in order to make the B2B area working. These mapping tasks arise at several levels[2] (cf. [16] for an overview):

- Different standards for describing products, or content standards (e.g., UN/SPSC[3] versus ecl@ss[4]).
- Different standards for describing the structure of the product catalogues, which contain the links to the content standards (e.g., Ariba[5] versus CommerceOne[6]).
- Different standards for describing exchanged business documents, such as purchase orders (e.g., XML Common Business Library xCBL[7] versus Commerce XML cXML[8]).

The first type of mappings mainly involves the real-world *semantics* of the information exchanged. They appear due to users who describe the same products in different ways and with different product classification schemes. The second and the third types of mappings arise in connection with the *syntactical structure* of the

[1] www.unece.org/trade/untdid/welcome.htm

[2] Not to mention the problem of mapping XML-based standards for updating product catalogues.

[3] www.unspsc.org

[4] www.eclass.de

[5] www.ariba.com

[6] www.commerceone.com/solutions/business/content.html

[7] www.xcbl.org

[8] www.cxml.org

information exchanged. The first mapping is usually provided by a content management solution provider. The second and the third mappings are usually provided by the B2B marketplace itself. From the technical point of view, the second and the third tasks are a kind of a document integration task, where the catalogs and exchanged documents must be translated and linked together. Most of the participants use XML to encode their documents, and a number of XML-based product description standards have been developed (cf. [6], [12]). There are around ten leading document standards for the B2B area existing now, but this number may increase further in the near future. Non-XML standards, such as EDIFACT or ISO STEP [9], are still in use. However, the corresponding XML encoding for them has been already developed and is now in the process of standardization. Hence, non-XML integration is unlikely to draw major interest in the future.

Some of these XML standards are compared by Li [12], who discusses seven different product description standards used in e-commerce, their complexity and potential integration problems. We will discuss this topic more extensively, proposing our architecture able to solve the integration problems highlighted in [12].

The XML transformation language XSL-T [3], together with the correspondent expression language XPath [4], provides the means to translate various XML documents. However, attempts to define the integration rules directly in XSL-T have revealed several important problems that make development of real-life integration rules very complicated [15]. Aside from the fact that XSL-T provides a low level of service for defining such mappings, the single-layer integration is a conceptual mismatch that poses serious obstacles to a direct mapping approach. Complex rules need to be defined to extract the semantic information from various syntactical styles, which translate the information at a semantic level, and which represent this information in a different syntax. Such rules are difficult to write and maintain; their re-usage is very limited. We have sought the solution to these problems in a layered approach for the catalog integration task. We distinguish between different sub-tasks in the overall mapping process, which enable identification of simple and re-usable rule patterns. Complex transformations are reached thought concatenation of a number of simple transformation rules.

In this paper, we discuss a layered model for business document integration and take the integration of address descriptions as a running example discussed in Section 2. Section 3 sketches the problems that arise in direct single-layered catalog integration. Section 4 introduces the three-layered model for the catalogs which consist of a *Syntax layer*, a *Data Models layer*, and an *Ontology layer*. Section 5 discusses the two-layered catalog integration approach. The paper ends with conclusions and future research directions.

2 The Running Example of XML Catalogs

We use the problem of *address* integration as a running example throughout the paper. An address is a simple business concept. It occurs very frequently in e-commerce and is an important part of any B2B mediation system. Unlike most of products, the structure of an address and the meaning of its components are understandable to everybody, which makes the explanation clear. At the same time, the alignment of various address representations of an address provides all major types of problems, which can appear in the product integration task.

The first standard analyzed in the paper is **xCBL** 3.0 developed by Commerce One[9], Inc. It provides a comprehensive set of standardized XML document formats, allowing buyers, suppliers, and service providers to integrate their existing systems into the electronic marketplaces [5]. The second standard is **cXML** 1.0 developed by a large consortium of companies including Ariba and Microsoft. cXML has been proposed for a similar purpose as xCBL, and it also targets document integration for the B2B mediation task. The DTDs for the address representation according to the above-mentioned two standards are presented in Fig. 1 (a)-(b) correspondingly.

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

(a) xCBL

```
<!ELEMENT PostalAddress (DeliverTo*, Street+, City, State?, PostalCode?, Country)>
```

(b) cXML

Fig. 1. The DTDs for an address

The representations of the same concept, the address, differ in each catalog. Conceptually equal document properties (e.g. denoting a street name) can be encoded with XML elements of different names; XML elements with the same names can have different meanings (especially this refers to the elements with unclear meaning, like Type), and ordering which is important in XML. Finally, the descriptions may have different granularity levels as required by the domain area, and provide or omit additional details. An example of the same address encoded with xCBL and cXML is shown in Fig. 2.

3 The Problems of the Single-Layered Integration

In the simplest case, we can integrate two XML catalogs directly by defining a set of XSL-T rules, as discussed in [15]. The rules directly translate each element or attribute of the first catalog into an appropriate XML element of the second catalog. See Fig. 3 (XSL-T rules, which directly map the cXML address to the xCBL format).

[9] www.commerceone.com

```xml
<OrganizationAddress>
    <ExternalAddressID>001</ExternalAddressID>
    <POBox/>
    <Street>De Boelelaan</Street>
    <HouseNumber>1081a</HouseNumber>
    <StreetSupplement1/>
    <StreetSupplement2/>
    <PostalCode>1081 hv</PostalCode>
    <City>Amsterdam</City>
    <Country>Netherlands</Country>
    <Region>North Holland</Region>
    <District/>
    <County/>
    <TradingPartnerTimezone/>
</OrganizationAddress>
```

(a) xCBL

```xml
<Address>
    <Name xml:lang="en">VU</Name>
    <PostalAddress name="VU">
    <DeliverTo>B. Omelayenko</DeliverTo>
    <Street>De Boelelaan, 1081a</Street>
    <City>Amsterdam</City>
    <State/>
    <PostalCode>1081 hv</PostalCode>
    <Country
isoCountryCode="NL">Netherlands</Country>
    </PostalAddress>
</Address>
```

(b) cXML

Fig. 2. The xCBL and cXML catalog examples

This approach mixes several independent tasks in a single XSL-T rule as shown in Fig. 3:

- Aligning different terminologies, e.g. mapping the xCBL element OrganizationAddress and the cXML element PostalAddress.
- Aligning the granularity level of the representations and performing necessary attribute splits with XPath expressions. For example, the Street cXML element, which actually refers to an address line with the street name and house number information, must be split into two separate Street and HouseNumber elements. Very often, this splitting is guided by ad-hoc rules, which make splits based on the element values.
- Transforming the attribute values.
- Restoring necessary syntactic formatting according to the target document standard.

The rules, which try to carry out the complete transformation process in one shot, have proven to be very complex. This causes serious problems in implementing and maintaining them. These problems are due to the mixture of several different aspects of the overall transformation process. For this reason, any re-use of such rules is practically impossible. Moreover, defining these rules from the scratch requires a great deal of manual effort. To overcome this bottleneck we have developed a multi-layered framework, which differentiates several aspects of the transformation process. Thus, we have been able to provide an approach where this translation is achieved by selecting and concatenating simple transformation rules. In short, we are able to transform a complex programming task into a simple plug-and-play process where the straightforward rule patterns are selected, instantiated, and combined. This multi-layered integration is discussed in the next section.

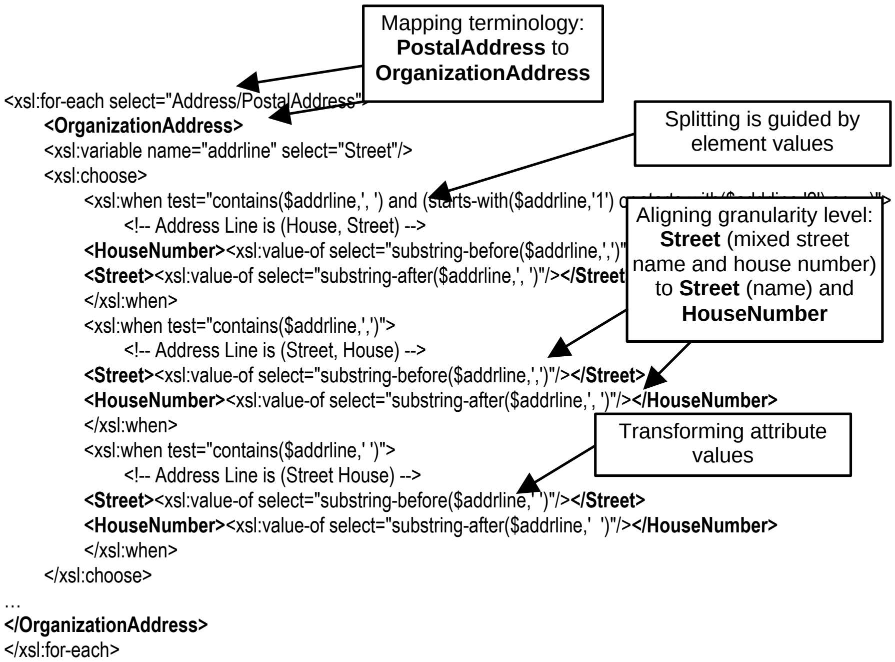

Fig. 3. Direct translation of the cXML address to the xCBL format. The elements, which form the resulting document, are presented in a bold font.

4 The Multi-Layered Model for Catalog Integration

The problems of single-layer integration occur because two tasks run together with a single set of transformation rules: syntactical translations between different XML representations and semantic mapping between the terminology and granularity level of the representations. Naturally, these two types of transformations belong to different layers of representation.

The layered approach for information representation on the Web was proposed in [13], where three layers, a syntax layer, an object layer, and a semantic layer are proposed for information modeling on the Web. The syntax layer provides a way of serializing information content into a sequence of characters according to some standard, e.g. XML. The purpose of the object layer is to offer an object-oriented view of the information with the normalized data models of standardized triples. Finally, the semantic layer provides a conceptual model for the information. We have based our integration architecture on this partitioning.

Hence, we separate three layers for the catalog integration task as presented in Fig. 4. These include the *Syntax layer*, the *Data Models layer*, and the *Ontology layer*.

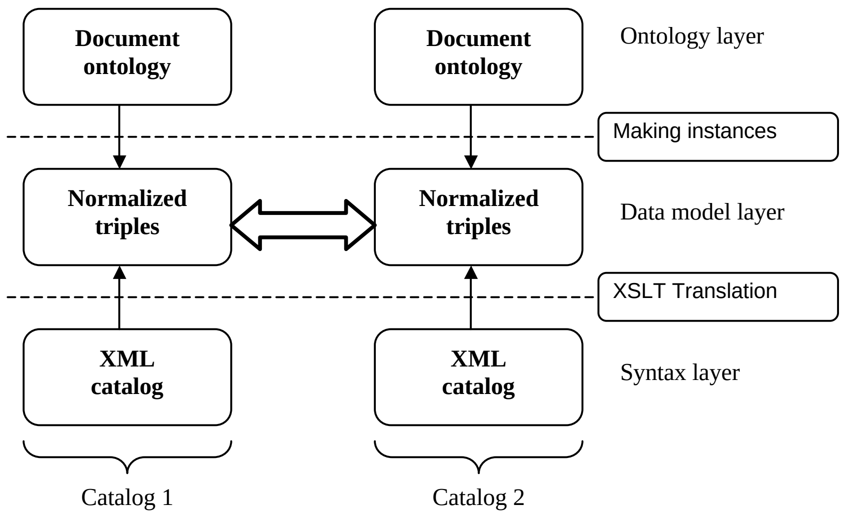

Fig. 4. The model for catalog information

The Syntax layer corresponds to the instance documents represented with their XML serialization. The serialization specifies used XML elements and attributes, and their order. Even semantically equal documents may differ in their serialization.

The Data models layer serves as a bridge between the Ontology layer and the Syntax layer. On this layer, the representations are abstracted from the differences imposed by the Syntax layer and the products are represented by object-property-value triples, where the properties stand for document elements.[10] Normalization is done according to the corresponding ontology, which specifies, for example, that the Street cXML element actually represents the street and house number information and must be split accordingly.

The terminology used at this layer is defined by the corresponding ontology and generally has to coincide with the terminology used at the Syntax layer. However, the former might be more detailed than the latter, e.g. the XML serialization may allocate only one element for the street name and house number, while the ontology has to allocate two separate elements. The lowest detail level also binds the quality of the mapping of a pair of catalogs. This problem emerges when we need to incorporate a new catalog into a marketplace. We cannot map it without information lost if the new catalog provides more details than the ontologies already used in the marketplace. Hence, we should always assume that the ontology provides the most detailed partitioning possible. According to the ontology, the data model must maintain these details even if they are not currently needed, as they may be required later in mapping a new catalog.

---

[10] These object-property-value triples may also be used to represent instance-SubclassOf-class triples according to the ontology defined at the ontology layer.

We assume that different terminologies must be aligned at the Ontology layer rather than at the Data Models layer. We will discuss this more extensively in upcoming papers as a future enhancement of the two-layered approach discussed here.

In our approach, we used RDF [11] on the Data Models layer as the language to encode the triples. RDF is a W3C[11] standard for describing machine-processable semantics of data that is also represented by the object-property-value triples. However, RDF is not the only language available for this layer. Another suitable candidate is Simple Object Access Protocol (SOAP)[12] (see [8] for a comparison of SOAP with RDF).

The Ontology layer is presented at the top of Fig. 4. It defines a terminology used to represent the information provided by various product and document standards. This layer specifies the terminology in detail, sufficient to define the transformations between the catalogs with one-to-one mapping rules, as shown in the next section.

In addition, the ontology contains the elements specified as optional and possibly absent in the XML serialization and, therefore, helps in aligning them. Although we occasionally refer to this layer throughout the paper, we do not present any further discussion of the possible ontology mismatches or integration problems, which may arise in this layer (see [10] for a relevant discussion). In this paper, we refer to it only as a pre-requisite in defining atomic elements for describing the pieces of exchanged information.

5 Two-Layered Information Integration

As mentioned earlier, simultaneous execution of several integration tasks causes the difficulties of the single-layered integration model. For this reason, we have used a 'divide-and-conquer' approach to decompose these tasks into several sub-tasks, each of which is performed separately.

The decomposition is performed in a similar way to the structure of heuristic classification proposed in [2]. Heuristic classification assumes that the classification is performed on a layer of abstract structures, and the input data must be first abstracted, i.e. translated from some particular format into abstract solution classes; after the classification, this intermediate solution must be refined to specific solutions.

The integration involves two layers: the Syntax layer and the Data Models layer. RDF documents have a standard XML serialization, and we treat the representations from both layers as XML documents and use XSL-T rules to transform them. However, there are many standard RDF (XML) serializations for the same set of triples. Even more, RDF is much more than an XML serialization for triples (and even not the best choice for a serialization). Enriched with RDF Schemas[13] which specify the structure of RDF triples, RDF Data Models layer must be regarded as a *modeling*

---

[11] www.w3c.org

[12] www.w3.org/TR/SOAP/

[13] www.w3.org/TR/2000/CR-rdf-schema-20000327/

layer, rather than an intermediate layer. Hence, 'pure' RDF-based technologies must be used at the Data Models layer.

In both layers, information is presented in XML, because RDF documents from the Data Models layer are serialized in XML; hence it is natural to use XSL-T language to define the transformation rules.

Translation of a catalog requires three steps as depicted in Fig. 5: (1) mapping the catalogs into their data models during the *abstraction* step; (2) mapping between the data models during the *transformation* step; (3) translating the new model into the target XML format during the *refinement* step. These three steps are described in the following subsections.

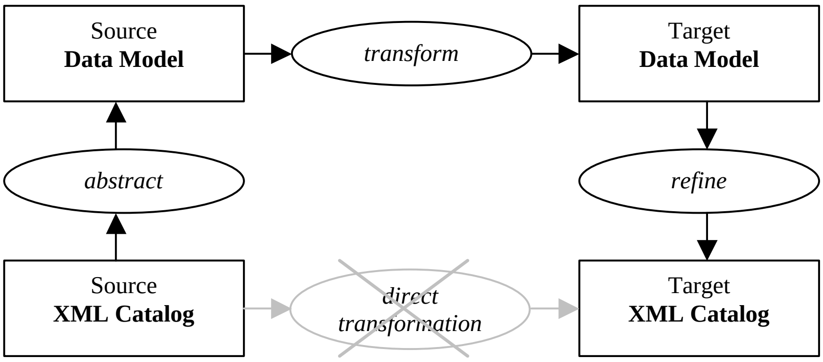

Fig. 5. The model for data transformation

Abstraction Step

At the abstraction step, the XML catalogs are translated into their data model encoded with RDF triples. This requires the following transformations:

- Translation of each XML element or XML attribute, which refers to a product feature into an RDF property with the same name.
- Split of a single XML element into two or more RDF triples, as specified by the corresponding ontology.
- Inclusion optional XML elements in the RDF triples, as specified by the ontology; the values of the elements are filled with the default value also specified in the ontology.
- Concatenation of multi-file descriptions into a single file.

At this stage, the transformations remain in the same or a more detailed terminology as specified by the input catalog. Hence, the abstraction step consists only of *one-to-many* (including *one-to-one*) mappings. The abstraction rules attempt to align the catalog to a more detail representation, hence, no *many-to-one* rules may appear at this atep.

One-to-one transformation is the simplest and most common type of transformation of an XML element into the corresponding RDF triple. Actually, it performs a syntactical translation between the XML and RDF serializations of the same element. For example, the following rule translates the cXML element PostalAddress and PostalCode to the xCBL elements OrganizationAddress and PostalCode:

```
<xsl:element name="rdf:Description">
 <xsl:for-each select="PostalAddress">
      <OrganizationAddress>
      …
      <PostalCode><xsl:value-of select="PostalCode"/></PostalCode>
      </OrganizationAddress>
      </xsl:for-each>
</xsl:element>
```

One-to-many mapping occurs when the ontology specifies several elements, which are represented with a single element in the XML serialization. The XSL-T language provides the means to represent mapping XML elements and attributes, as well as possibilities to analyze text inside an element and to split it into two or more. XSL-T uses the XPath language to perform these operations. Accordingly, XSL-T rules must be extended with small XPath expressions (element parsers) which perform necessary element splits. For example, in the following fragment of an cXML address, it is assumed that the element Street contains street name separated from the following house number with a comma:

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

We must split Street into two XML elements

```
<Street>De Boelelaan</Street>
<HouseNumber>1081a</HouseNumber>
```

as specified in the cXML ontology. This can be done with the following XSL-T rule:

```
<xsl:element name="rdf:Description">
     <OrganizationAddress>
          <xsl:variable name="addrline" select="Street"/>
          <Street><xsl:value-of select="substring-before($addrline,',')"/></Street>
          <HouseNumber><xsl:value-of select="substring-after($addrline,', ')"/></HouseNumber>
          …
     </OrganizationAddress>
</xsl:element>
```

The above example, as well as the example from Fig. 3, presents only a few ways to place a street name and a house number in a single line. For example, an assumption that the street name starts from a letter used in Fig. 3 would fail to recognize the house number in the line ‘5-th Avenue 5’. Additional information might also be needed. In consequence, our approach sub-divides this step into two sub-steps:

(1) transformation of the content presented by separate XML tags; here we can employ the semi-formal structure of the content; (2) wrapping natural language sequences and splitting them into separate information units; here we relay on the wrapper and information extraction technology applied to unstructured information sources.[14]

Transformation Step

All inter-catalog mappings are performed on the layer of RDF data models. We assume that the ontologies specify the catalogs on a detail level sufficient to specify all inter-catalog mappings with *one-to-one* transformations, where all necessary element splits are performed during the abstraction step (Fig. 5) and the necessary element merges will be done on the refinement step. Hence, only *one-to-one* mappings may appear on this step.

One-to-one mappings are done in the same way as at the abstraction step. At the transformation step, the mappings actually perform syntactical transformations between the RDF serializations of the data models of the catalogs. The transformation rules may appear as follows (from cXML to xCBL):

```
<xsl:when test="@about='cXML'">
    <xsl:attribute name="about">xCBL</xsl:attribute>
        <OrganizationAddress><xsl:value-of select="PostalAddress"/></OrganizationAddress>
        …
</xsl:when>
```

Refinement Step

During the refinement step, all syntactical restrictions required by the target format are restored, and necessary many-to-one transformations are performed:

- Each RDF triple is serialized with a corresponding XML element or attribute.
- One or more RDF triples have to be merged into a single XML element, if required.
- Target XML elements are created in proper order.
- The target XML representation may be partitioned into several files, if required.

In consequence, only *many-to-one* (including *one-to-one*) rules may appear at this step. Both types of rules can be easily implemented with XSL-T without XPath fragments.

One-to-one mappings are done in a straightforward way with XSL-T rules which transfer the RDF triples into the XML catalog elements and attributes. In the example shown in Fig. 6, the PostalCode element is translated into XML with the one-to-one mapping.

---

[14] See *RISE: Repository of Online Information Sources Used in Information Extraction Tasks*: http://www.isi.edu/~muslea/RISE/ for a survey on the available technology in this area.

```
<xsl:for-each select="rdf:Description">
 <PostalAddress name="default">
     <DeliverTo><xsl:value-of select="PersonName"/></DeliverTo>
     <Street>
         <xsl:value-of select="Street"/>, <xsl:value-of select="HouseNumber"/>
     </Street>
     <PostalCode>
         <xsl:value-of select="PostalCode"/>
     </PostalCode>
 </PostalAddress>
</xsl:for-each>
```

One-to-Many mapping

One-to-One mapping

Fig. 6. Refinement rules for cXML

Many-to-one mapping occurs when two or more RDF attributes must be translated to one XML element or attribute. For example, the cXML ontology stores the street name and the house number separately for simplicity of inter-catalog mappings discussed earlier. However, the XML serialization assumes that both elements must be split into a single Street element. This is done by means of XSL-T, merging the elements

```
<Street>De Boelelaan</Street>
<HouseNumber>1081a</HouseNumber>
```

into the following cXML element:

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

as shown in Fig. 6, Street creation rule.

6 Conclusions

In this paper, we sketched the problems, which occur in the catalog integration task when approached directly without intermediate sub-steps relaying on a multi-layered approach (see [15] for more details). We presented a two-layer integration approach (implicitly referring to a third layer, which we will discuss in the upcoming papers). The introduction of the second layer makes it possible to decompose the integration task into three simpler tasks: *abstraction*, *transformation*, and *refinement*. Each of the tasks can be resolved and debugged separately. Up to a certain degree, this approach overcomes the main problem of the information integration task in B2B electronic commerce. The problem lies in the high complexity and unreadability of the rules, which still have to be developed manually. In our approach, such translation can be achieved by selecting, instantiating, and combining elementary translation rule patterns.

At present, we are working on a framework and tool environment, which allows effective and efficient definition of such mappings. This framework must provide:

- A simple language on top of XSLT, which is customized to the specific needs of mapping rules in electronic commerce. Instead of manually defining transformation directly in XSLT, they should be derivable from the mappings defined at a more intuitive level. This helps to transform a complex programming task into a simple plug-and-play process based on the simplified rule patterns, identified by separating different mapping aspects.
- We used XSLT rules to translate the data models throughout the paper. The data models layer is created to hold ontology instances. They are encoded with RDF triples and must conform an ontology (schema), most likely represented with RDF Schema. Future development of the architecture requires dealing with RDF and RDF Schema querying, verification of RDF to RDF Schema, and performing inference on the Schemas. We will examine possible ways to exploit the Sesame[15] tool (cf. [1] for a state of the art report) for this process.
- Finally, the integration architecture will consist of several ontologies, each of which will specify a certain aspect of B2B mediation. They are: content standards, content aligning ontologies, document ontologies (which instances from the data model discussed in this paper), partner codes (e.g., UDDI[16]), and a workflow ontology specifying the way how all other ontologies must interact together to make a B2B marketplace functioning.

Successful B2B electronic commerce must deal with three serious mapping problems: (1) different content standards define (in different ways) over 10,000 classes and ten times more attributes for describing products; (2) different document standards define (in different ways) over 400 business documents exchanged in electronic trading; (3) different product catalog standards define (also in different ways) complex structures for describing the products exchanged. In short, B2B marketplaces face a need for the development of an appropriate technology, which allows them to easily define their large number of complicated mappings. Otherwise, they will suffer the same fate that left the tower of Babel an unfinished ruin.

Acknowledgements. We would like to thank Hans Akkermans, Guy Botquin, Ying Ding, Stefan Decker, Michel Klein, Ellen Schulten, Volodymyr Zykov, and three anonymous referees for their insightful comments, which helped us in writing this paper.

References

[1] Broekstra, J., Fluit, C., van Harmelen, F.: The State of the Art on Representation and Query Languages for Semistructured Data. IST-1999-10132 On-To-Knowledge Project, Deliverable 8, August (2000); available online at: http://www.ontoknowledge.org/del.shtml

[2] Clancey, W.: Heuristic Classification, Artificial Intelligence **27** (1985) 289-351

---

[15] http://sesame.aidministrator.nl/

[16] www.uddi.org

[3] Clark, J.: XSL Transformations (XSL-T), W3C Recommendation, November (1999); available online at http://www.w3.org/TR/xslt/
[4] Clark, J., DeRose, S.: XML Path Language (XPath), version 1.0, W3C Recommendation, November 16 (1999); available online at http://www.w3.org/TR/xpath
[5] Commerce One, Inc.: Commerce One XML Common Business Library (xCBL) 3.0, Press release made at the eLink Conference, Hong Kong, November 29 (2000); available online at http://www.commerceone.com/news/us/xcbl30.html
[6] Fensel, D.: Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag, Berlin (2001)
[7] Fensel, D., van Harmelen, F., Akkermans, H., Klein, M., Broekstra, J., Fluit, C., van der Meer, J., Schnurr, H.-P., Studer, R., Davies, J., Hughes, J., Krohn, U., Engels, R., Bremdahl, B., Ygge, F., Reimer, U., Horrocks, I.: OnToKnowledge: Ontology-based Tools for Knowledge Management, In: Proceedings of the eBusiness and eWork 2000 Conference (EMMSEC-2000), Madrid, Spain, October 18-20 (2000)
[8] Haustein, S.: Semantic Web Languages: RDF vs. SOAP Serialization, In: Proceedings of the Workshop on the Semantic Web - SemWeb'2001 at the 10-th WWW Conference, Hong Kong, May 1 (2001)
[9] Integrated generic resource: Fundamentals of product description and support, International Standard ISO 10303-41, Second Edition (2000)
[10] Klein, M.: Combining and relating ontologies: an analysis of problems and solutions. In: Gomez-Perez, A., Gruninger, M., Stuckenschmidt, H., Uschold, M. (eds.): Proceedings of the Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, USA, August (2001); available online at http://www.cs.vu.nl/~mcaklein/
[11] Lassila, O., Swick, R.: Resource Description Frame-work (RDF) Model and Syntax Specification, W3C Recommendation, February (1999); available online at http://www.w3.org/TR/REC-rdf-syntax/
[12] Li, H.: XML and Industrial Standards for Electronic Commerce, Knowledge and Information Systems **2** (2000) 487-497
[13] Melnik, S., Decker, S.: A Layered Approach to Information Modeling and Interoperability on the Web, In: Proceedings of the Workshop on the Semantic Web at the Fourth European Conference on Research and Advanced Technology for Digital Libraries (ECDL-2000), Lisbon, Portugal, September 21 (2000)
[14] Open Applications Group Inc.: Open Applications Group Integration Specification, OAGIS Release 7.0.2 (2000); available online at http://www.openapplications.org/
[15] Omelayenko, B., Fensel, D.: An Analysis of the Integration Problems of XML-Based Catalogues for B2B Electronic Commerce, In: Proceedings of 9th IFIP 2.6 Working Conference on Database Semantics, Hong Kong, April 25-28 (2001); available online at http://www.cs.vu.nl/~borys/papers/
[16] Omelayenko, B., Fensel, D.: An Analysis of B2B Catalogue Integration problems: Content and Document Integration, In: Proceedings of the International Conference on Enterprise Information Systems (ICEIS-2001), Setúbal, Portugal, July (2001); available online at http://www.cs.vu.nl/~borys/papers/
[17] U.S. Department of Commerce: Digital Economy 2000, White paper, June (2000)