Adapting RosettaNet B2B Standard to Semantic Web Technologies

Jamel Eddine Jridi and Guy Lapalme
Université de Montréal, Department of Computer Science and Operations Research, Montréal, Québec, Canada

Keywords: Business Ontology, Semantic Web, Business Process, Standards Transformation, RosettaNet, B2B.

Abstract: This paper presents a methodology for adapting RosettaNet B2B Standard to Semantic Web technologies. We present an approach for mapping RosettaNet Partner Interface Process (PIP) descriptions defined currently with DTD or XML Schemas format to an ontological representation using an OWL/XML rendering. It has been applied to the full set of PIPs. The transfer is transparent and reaps the benefits of the formalization effort put in the XML representation. All produced RosettaNet PIPs ontologies were validated according to the XML Schema of OWL/XML Serialization and their consistency checked with a OWL Reasoner. The use of formal semantics for B2B based on ontologies ensures a better interoperability and should help overcome the problem of standards integration by the use of ontology alignment.

1 INTRODUCTION

Most business transactions are now made via the web and managed by information exchanges between organizations. These exchanges are becoming more and more complex, which makes interoperability and data integration difficult. In order to communicate fruitfully between partners, it is necessary that each partner uses the same definition of the information. It is crucial to determine how information and data are defined and used to design an abstract and common model describing the structure and semantics of concepts. Many organisations, like OASIS and UN/EDIFACT, have tried to build common business models and languages based on analysis and design methods.

Several business to business (B2B) standards are XML based. XML Schema services provide the flexibility to create, manipulate, model, organize and share information, but only from a syntactical point of view. They do not deal with the semantics of the data. Research is now envisioning the adoption of semantic web technologies in the business domain. The question being: *What is needed from the semantic web technologies?*

(Lytras and García, 2008) argue for the need of semantic web in the business domain and analyse the practical requirements in terms of interoperability or knowledge representation. The use of formal semantics for B2B based on ontologies should help overcome the problem of standards integration by the use of ontology alignement. The use of ontologies is not only for communication between different applications but also to provide reasoning support to infer, integrate information and to extract meaning. Most existing approaches, such as RosettaNet[1] and ebXML[2], are still XML based, but their proposed ontology conterparts are still unrelated to their XML representation.

In this paper, we propose an automatic transformation and mapping of existing B2B standards based on DTD or XML Schema to an ontological representation based on semantic web techniques using an OWL/XML rendering. Our goal is to make the transfer more transparent and reap the benefits of the effort put in the development of XML business standards. We propose a novel methodology to adapt all RosettaNet PIPs, in XML Schema and DTD format, to semantic web technologies using the OWL language. Existing approaches to ontologise RosettaNet (Haller et al., 2008) targetted web service languages like WSMLwhich are not W3C recommendations yet. Other approaches, targetted to OWL a W3C recommandation, propose a handmade ontology unrelated to the original XML representation.

The remainder of this paper is organized as follows: Section 2 gives a short historical overview of B2B standards and electronic data interchange (EDI) evolution for which we discuss the limits and benefits.

[1] http://www.rosettanet.org/
[2] http://www.ebxml.org/

Eddine Jridi J. and Lapalme G..
Adapting RosettaNet B2B Standard to Semantic Web Technologies.
DOI: 10.5220/0004434404430450
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 443-450
ISBN: 978-989-8565-60-0

In Section 3, we describe our methodology for mapping from DTD, XML Schema to OWL that has been applied to all RosettaNet PIPs. In Section 4, a case study, the processing of `PIP3A4 Request Purchase Order`, will be described in more details. Section 5 concludes the paper and presents some future work.

2 STATE OF THE ART

XML, which identifies data by tagging information, has been used for defining formats for data exchange between applications of e-business. xCBL[3], RosettaNet (RosettaNet Program Office, 2011) and ebXML (Huemer, 2002) have been proposed for the management of business documents using XML technologies. Their goal is to build a common business model between partners.

ebXML, trying to facilitate e-commerce for all businesses, provides meta-models describing business processes (Kantola and Korhonen, 2002), but their description are quite general and don't provide a specific definition of business processes, thus it can't reduce the uncertainty and confusion.

On the other hand, RosettaNet defines processes, business documents and messaging in details. But it focusses on a specific segment of the industry e.g. the supply chain.

But these XML based standards do not deal with the semantics of information. For this, we not only need to store document metadata (e.g. author, title, etc.), but we must also make available to the machines the most important concepts found in the document, the relations between these concepts or that of other documents, etc. By using semantic models based on ontologies, companies acquire several benefits such as the ability to perform inference on knowledge bases and the capacity to share domain models to easily exchange and integrate information (Cardoso and Bussler, 2011).

Research now focuses on the adoption of semantics for the business domain. Several formal languages have been proposed: WSMO (Roman et al., 2005), and OWL-S (W3C Member Submission, a). These web service languages are still in W3C submission member stage and are not recommended yet. (Cardoso and Bussler, 2011) say that these initiatives are somewhat limited when transposed to real-world industries settings and thus may be considered a penalty.

B2B standards have been the subject of many studies for adding semantics. (Dogac et al., 2004) presents an extension to ebXML registry with an OWL ontology but it's difficult to apply reasoning on it or to verify its consistency. In spite of the adaptation effort, much work is still to be done especially on reasoners. (Kotinurmi et al., 2011) proposed an "ontologically-enhanced RosettaNet" efforts to map RosettaNet business documents into a Web ontology language, allowing business reasoning based on RosettaNet message exchanges. They dealt with business process alignment and ambiguous message definitions.They demonstrate their solution using WSMO (Roman et al., 2005) and the properties of WSML language (W3C Member Submission, b). On other hand, (Haller et al., 2008) propose a transformation methodology of RosettaNet PIPs available as XML Schemas to WSML. Without giving implementation details, they dealt only with 50 RosettaNet PIPs defined as XML Schemas, but not with those available as DTD format.

WSML is limited to Web services only and never passed the W3C member submission stage, while OWL is a general-purpose ontology language for the semantic Web as a W3C recommendation (Panziera et al., 2010). OWL can provide a better interoperability, a support for ontology reuse, and the possibility to exploit several mature matching techniques (Panziera et al., 2010). An ontology can help identify basic concepts describing a knowledge domain and define a common vocabulary for organizing documents.

There have been some ontology based attempts: eCl@ssOWL[4] ontology based on services classification and products description; another important work was RosettaNet Ontology[5], an handmade OWL representation of PIP3A4 RosettaNet Partner Interface Processes (PIPs), which lacks many details and relies on properties not defined in the XML representation like `has_price_n_availability_line_item`. This ontology does not solve the problem of uncertainty and ambiguity. So, corporations using RosettaNet must spend more effort to adopt the new terms defined in the RosettaNet Ontology.

For this reason, it crucial to start from existing documents based on XML Schemas in order to keep the same list of concepts and terms defined.

Our approach is based on an automatic mapping from existing DTD and XML Schema representations to OWL ontologies using the OWL/XML serialization. Some approaches similar to ours have been proposed: Gloze (Battle, 2006), XS2OWL[6] and Re-

[3]XML Common Business Library, http://www.xcbl.org/

[4]http://www.heppnetz.de/projects/eclassowl/

[5]http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/ontologies/rosetta.owl

[6]http://www.music.tuc.gr/projects/sw/xs2owl/

DeFer[7] projects. (Battle, 2006) presents a simple translation between XML and RDF based on XML Schema but without representing the semantics defined by the XML Schema by an ontology. ReDeFer overcomes this limitation and represent an "XML Semantic Reuse Methodology" that combines an XML Schema to OWL mapping, named XS2OWL, with an XML to RDF mapping named XML2RDF (García and Gil, 2007). But, some details are missing e.g. properties having enumerated values are not translated by ReDeFer and XSD annotations are not dealt with. OntologyDesignPatterns.org[8], a Semantic Web portal dedicated to ontology design patterns, describes experiences and problems encountered with ReDefer tool. (Tsinaraki and Christodoulakis, 2007) pinpoints that some XML Schema construct transformations to OWL in ReDeFer do not follow closely the XML Schema semantics.

As discussed in (García and Gil, 2007) and (Tsinaraki and Christodoulakis, 2007), the ReDeFer XML2OWL is used to map one of the main B2B standards like ebXML, but XS2OWL proposes a similar approach to map a multimedia standards like MPEG-7 and MPEG-21. We tested the RedeFer XML2OWL and XS2OWL (Tsinaraki and Christodoulakis, 2007) using the RosettaNet PIP3A4 (see section 4), but it resulted in an inconsistent ontology due to the complexity of PIPs Schema. Consistency was evaluated using the Pellet reasoner (Sirin et al., 2007). As RosettaNet is one of the most prominent standards in B2B, we now propose a transparent and direct mapping of PIPs.

3 OUR APPROACH

We now propose a mapping of DTD and XML Schema documents to an ontology using the OWL/XML serialization (Motik et al., 2012). We choose OWL/XML serialization syntax for the abstract OWL 2 Structural Specification because it closely models the structure of an ontology, it is more readable (Robinson and Bauer, 2011) and is easier to process using XML tools like XQuery. An XSLT stylesheet, named OWL XGX[9] can convert an OWL/XML serialization to the standard RDF/XML syntax.

Our approach achieves a complete mapping integrating the details missing in previous mapping approaches. To illustrate it, we will use the XML based RosettaNet PIPs that do not deal yet with the semantics of data.

3.1 What is RosettaNet?

The RosettaNet consortium provides a global forum for suppliers, customers, and competitors to do business and collaboration in an efficient and profitable manner. Since 1998, RosettaNet is used and endorsed by several corporations such as Intel, Oracle, Cisco and Microsoft. (Wang and Song, 2006) introduces a typical scenario and the benefits of using RosettaNet standards, they also survey the architectures supporting RosettaNet.

To manage business activities, RosettaNet formalizes Partner Interface Processes (PIP) with either Data Type Definition (DTD) format or XML Schema, and define business processes between trading partners. PIPs are organized into eight groups of core business processes called clusters, themselves further grouped into segments. Each segment includes several PIPs.

The RosettaNet architecture contains a Business Dictionary to define the properties for basic business activities and Technical Dictionaries to provide properties for products (Wang and Song, 2006). The RosettaNet Implementation Framework (RNIF) describes the packaging, routing, and transport of all PIP messages and business signals.

3.2 System Architecture

Our approach is based on different mapping rules to transfer XML Schema to OWL/XML rendering. To illustrate these rules, we will use examples drawn from RosettaNet PIPs documents.

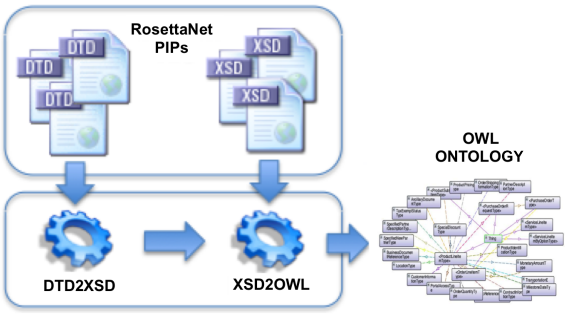

Figure 1: Overview of our system architecture.

An overview of our system architecture is sketched in Figure 1. The system core is composed of two parts: DTD2XSD and XSD2OWL. As input, it takes RosettaNet PIPS as either DTD or XSD documents and transforms them into an OWL ontology with XSLT 2 stylesheets. XSLT (Extensible

[7]ReDeFer, http://rhizomik.net/redefer

[8]http://ontologydesignpatterns.org/ont/fao/figis/speciesNTK.owl

[9]http://www.w3.org/2009/owl-xgx/

Stylesheet Language Transformations) is a language for transforming XML documents into other XML documents or other formats like HTML.

3.2.1 XSD2OWL

The main part of our work is the XSD2OWL Engine which offers a transparent and direct transformation of documents from an XML Schema to an OWL ontology based on OWL/XML rendering and OWL 2 language which adds new features compared to RDF/RDFS and OWL 1, e.g. data keys, data ranges, cardinality, asymmetric properties, reflexivity, etc (Hitzler et al., 2009).

Our XSD2OWL mapping is based on XSLT and XPath languages to capture and navigate the Schema. Every XPath expression refers to an XML element definition that appears in the document. Table 1 illustrates some mapping rules used in the transfer process.

Element names and namespaces defined in the XML Schema are preserved in the OWL representation. In the case of `owl:DataProperty` and `owl:ObjectProperty` in order to improve readability, a prefix is added to the name of a property, e.g. `cityName` will become `has_cityName`. We now describe in more detail the rules illustrated in Table 1.

RULE 1. OWL classes represent individuals having common features. XML Schema Complex Type describes XML instances with common properties. Also, when XPath pointers (`xsd:element/xsd:complexType`) refers to a complex element having a child. In these cases, they will be mapped to an OWL class.

RULE 2. An XML Schema element with attributes is considered a complex type. XML Schema elements are transformed to OWL properties (`owl:ObjectProperty` or `owl:DataProperty`) having as `owl:range` the `@type` attribute value. If the value refers to an element belonging to an external namespace (e.g. `uuom:UnitOfMeasure`) and having a prefix different to the target namespace (`tns:`). Its value determines the type of the OWL property:

- If it is a simple XML element, with a `@type` attribute, or a type deriving from `anySpecialType` in the XML Schema. It is transformed to an `owl:DataProperty` (node to value relations).
- If the value refers to a complex type, an `owl:ObjectProperty` is built (node to node relations).

RULE 3. An XML Schema attribute is always declared as a simple type where `@name` is the name of the attribute and `@type` specifies his data type. So, as shown in Table 1, all `xsd:attribute` are transferred to OWL datatype properties (except that `schemaVersion` attribute are removed). RosettaNet provides an attribute `schemaVersion` to identify the versions of the schema.

RULE 4. An annotation element is a schema comment and serves as documentation. In RosettaNet, annotations include the PIP name, version, role and elements definition. However, all "`xsd:annotation//urss:definition`" are mapped to `AnnotationAssertion` having `rdfs:comment` as `AnnotationProperty`.

RULE 5. Occurrence indicators are used to define how often an element can occur. The `@maxOccurs` and `@minOccurs` indicators specify the maximum and minimum number of element occurrences. They are mapped to `AnnotationAssertion` having `owl:maxCardinality` and `owl:minCardinality` as `AnnotationProperty`.

RULE 6. The simple Type element specifies the constraints and information about the values of attributes or text-only elements. As defined in RULE 1, a simple element is an XML element that contains a `@type` attribute of one of the XML Schema types or defined as a `xsd:simpleType`, and mapped to `owl:DataProperty`. The property range `owl:DataPropertyRange` takes the attribute `xsd:simpleType/xsd:restriction/@base` value.

RULE 7. Restrictions are used to define constraints on values for XML elements or attributes. Enumeration is one of the constraints used to limit the content of an XML element to a set of acceptable values. In OWL, the property ranges are mapped using `owl:DataOneOf` element that contains some `owl:literal` constructs to define the property values.

These rules has been implemented within the XSD2OWL XSLT stylesheet composed of 4 main templates. The `<xsl:template>` element defines rules to apply when a specified node is matched using XPath expressions. For this, we present a pseudo-code which describes the instructions and the rules to apply in one of the principal templates in XSD2OWL. The Element and complexType Templates are respectively presented in the algorithms 1 and 2 described here in pseudo-code.

The limits of our mapping process is that some OWL semantic relations are not dealt with `subClassOf`, `equivalentClass`, `subPropertyOf`, etc. In the future, we consider using ontology matching and alignment approach to overcome these limits.

The result of the application of these rules is an OWL ontology making the semantics of the XML Schema more explicit. Our transformation process transfers XML Schema constructs to OWL ones. To

Table 1: Examples of XSD2OWL Mapping Rules as produced by the XSLT templates. Numbers in the first column are refered to in Section 3.2.1.

N	XML Schema Construct	OWL 2 Construct
1	<xsd:complexType name="A"/> or <xsd:element name="A"/> <xsd:complexType/> </xsd:element>	<owl:Declaration> <owl:Class IRI="A"/> </owl:Declaration>
2	<xsd:element name="A" type=""/>	<owl:Declaration> <owl:DataProperty IRI="A"/> </owl:Declaration> or <owl:Declaration> <owl:ObjectProperty IRI="A"/> </owl:Declaration>
3	<xsd:attribute name="A" type=""/>	<owl:Declaration> <owl:DataProperty IRI="A"/> </owl:Declaration>
4	<xsd:element name="A" type=""> <xsd:annotation> <xsd:appinfo> <urss:Definition>B</urss:Definition> </xsd:appinfo> </xsd:annotation> </element>	<owl:AnnotationAssertion> <owl:AnnotationProperty IRI="&rdfs;comment"/> <owl:IRI>A</owl:IRI> <owl:Literal datatypeIRI="&rdf;PlainLiteral">B</owl:Literal> </owl:AnnotationAssertion>
5	<xsd:element name="A" minOccurs="0"/> <xsd:element name="B" maxOccurs="2"/>	<owl:AnnotationAssertion> <owl:AnnotationProperty abbreviatedIRI="owl:minCardinality"/> <owl:IRI>A</owl:IRI> <owl:Literal datatypeIRI="&xsd;nonNegativeInteger">0</owl:Literal> </owl:AnnotationAssertion> <owl:AnnotationAssertion> <owl:AnnotationProperty abbreviatedIRI="owl:maxCardinality"/> <owl:IRI>B</owl:IRI> <owl:Literal datatypeIRI="&xsd;nonNegativeInteger">2</owl:Literal> </owl:AnnotationAssertion>
6	<xsd:element name="A" type="xsd:string"/> or <xsd:element name="A" type="B"/> <xsd:simpleType name="B"> <xsd:restriction base="xsd:string"/> </xsd:simpleType>	<owl:DataPropertyRange> <owl:DataProperty IRI="A"/> <owl:Datatype IRI="&xsd;string"/> </owl:DataPropertyRange>
7	<xsd:element name="A" type="B"/> <xsd:simpleType name="B"> <xsd:restriction base="xsd:token"> <xsd:enumeration value="OWL"/> <xsd:enumeration value="XML"/> </xsd:restriction> </xsd:simpleType>	<owl:DataPropertyRange> <owl:DataProperty IRI="A"/> <owl:DataUnionOf> <owl:DataOneOf> <owl:Literal datatypeIRI="&rdf;PlainLiteral">OWL</owl:Literal> <owl:Literal datatypeIRI="&rdf;PlainLiteral">XML</owl:Literal> </owl:DataOneOf> </owl:DataUnionOf> </owl:DataPropertyRange>

validate our results, we apply two types of validation (syntactical and semantic).

The syntactical validation consists of checking the conformity of our results with the XML schema for OWL/XML serialization[10].

[10]http://www.w3.org/TR/2012/REC-owl2-xml-serialization-20121211/

Algorithm 1: Element Template.

```
for each xsd:element visited do
  name ← null
  if xsd:element/@ref then
    name ← xsd : element/@ref
  end if
  if xsd:element/@name then
    name ← xsd : element/@name
  end if
  Current ← FIND_ELEMENT(name)
  type ← VERIFY@TYPE(Current/@type)
  if type refer to complexElement then
    APPLY RULE 2 (Object Property).
  end if
  if type refer to simpleElement or dataType then
    APPLY RULE 2 (Data Property).
    if type is SIMPLE or DataType then
      APPLY RULE 6 (DataPropertyRange).
    end if
    if type is ENUMERATION VALUES then
      APPLY RULE 7 (DataPropertyRange).
    end if
  end if
  APPLY RULE 5 (Cardinality).
  APPLY RULE 4 (AnnotationAssertion).
end for
```

Algorithm 2: complexType Template.

```
for each xsd:complexType visited do
  name ← xsd : complexType/@name
  APPLY RULE 1 (Class).
  APPLY RULE 4 (AnnotationAssertion).
  CALL Element Template.
  CALL Attribute Template.
end for
```

The semantic validation checks the consistency of the resulting ontologies. The consistency has been checked using OWL validators offered by `Protégé` tool, e.g `Pellet`. The use of ontologies is not only useful for communication between different applications but their reasoning support is important to extract information meaning and applying SPARQL semantic queries.

The mapping rules application have been applied on the 112 PIPs available for download from the RosettaNet website among the 132 PIPs on the PIP Directory.

3.2.2 DTD2XSD

Most PIPs are defined with XML Schema, but 22 of them use DTDs. So, we decided to use Trang[11] to transform these DTDs into an XML Schema before applying the stylesheet defined in the previous section. Trang is an existing open source program for converting between different schema languages, e.g. RELAX NG to XML Schema, or DTDs to XML Schema.

Trang output is a set of XSD files that require a cleaning step due to the complexity of the DTD files. We ensured by manual inspection that all examples conform to valid transformations. Then, XSD2OWL Mapping rules shown in the previous section are applied to the Trang output.

As shown in the section 3.2.1, the annotation and element description serve as documentation and understanding the element semantics. Within XML Schema documents, RosettaNet provide element description with annotation element (RULE 4). But RosettaNet DTD documents define semantics using an HTML document that contains an element explanation. So, for every element in the XSD2OWL mapping process, we parse the HMTL document to match the associated definition. Before parsing step, JTidy[12] is used for cleaning up malformed HTML.

The execution time taken for mapping RosettaNet PIPs to OWL representation is about 55 seconds on a machine having 64 MegaByte of internal memory. The processing time is the sum of the DTD2XSD time (19 sec) and XSD2OWL one (36 sec).

From syntactical point of view, the OWL documents of all RosettaNet PIPs are conform to the XML Schema for OWL/XML serialization.

4 CASE STUDY

This section illustrates the mapping process using a case study. The `RosettaNet PIP3A4`, version `V11.14.00` published on 9 February 2010.

We discuss the principles of PIP3A4 and describe the ontology resulting from the transformation process.

The `PIP3A4`, named `Request Purchase Order`, belongs to the `Order Management` cluster and `Quote and Order Entry` segment. It allows a buyer to issue a purchase order and get a seller confirmation. There are several interactions within this process, which starts by a sending purchase order, until the receipt of seller's confirmation, if the purchase order was accepted, rejected or delayed. So, RosettaNet provides an XML Schema that describes different concepts used such as : `PurchaseOrderRequest` (the root node) contains `DocumentHeader` that describes

[11] http://www.thaiopensource.com/relaxng/trang.html

[12] http://jtidy.sourceforge.net/

Table 2: Computational Statistics related to RosettaNet PIPs and PIP3A4 `Purchase Order Request`.

	Purchase Order Request PIP3A4	All PIPs
Number Of ComplexTypes	7	1251
Number Of Attributes	1	105
@type Refer to SimpleTypes	1	103
Number Of Element@ref	23	3051
Number Of Element@name	37	2676
Line Of Code in the original XSD Files	782	36494
Line Of Code in generated OWL Files	5565	66368
Time needed to transform XSD to OWL	300 milliseconds	36 sec

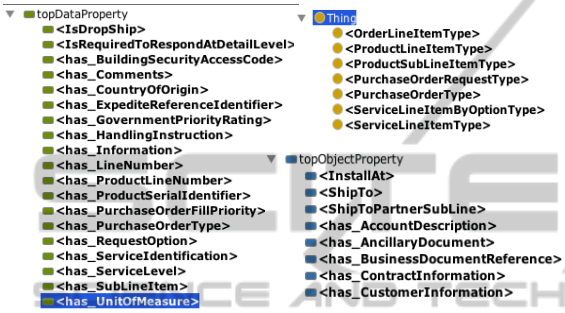

Figure 2: Screen prints tabs from `Protégé` describing the PIP3A4 Ontology generated by XSD2OWL Mapping (Classes, Data Properties and Object Properties).

document informations (creation date, version, etc.) and `PurchaseOrder` which presents the order specification (items description, partner identification, etc).

As RosettaNet PIPs are not in the public domain. We cannot publish the PIP3A4 but we will describe the resulting ontology with some statistics before and after the application of our mapping rules. Table 2 illustrates some computational statistics related to RosettaNet PIPs and the PIP3A4.

The resulting ontology contains 7 classes, 19 data properties and 41 object properties. The ontology IRI is `urn:rosettanet:specification:interchange:PurchaseOrderRequest:xsd:schema:02.05`. In the rest of this section, we will take one example from each OWL construct(Class, DataProperty and ObjectProperty).

Class. `<OrderLineItemType>` element defines the collection of business properties that describe an entry in a purchase order business document. This definition is taken from annotation comment. This element represents a complex type in the associated XML Schema document. The algorithm 2 has been executed to transform ComplexTypes by applying the RULE 1.

DataProperty. `<has_UnitOfMeasure>` represents a code to identify product unit measure. As shown in previous section, the algorithm 1 is used for each `xsd:element` (e.g.`UnitOfMeasure`). The RULE 2 has been applied to transform it to an `DataProperty`. In this case, the result of `VERIFY@TYPE()` function, having the `@type` attribute as parameter, refer to a `simpleType` having an enumeration values. So, the RULE 7 has been applied to specify the `DataPropertyRange`.

This property has as domain the `<ProductLineItemType>` class and as range a set of acceptable values like `10P:10-pack` and `1BF:100 Board Feet`.

ObjectProperty. `<ShipTo>` element refers to the partner and/or location to which the product must be delivered. This property links two classes. The property domain is `<ProductLineItemType>` and the property range is `<SpecifiedPartnerDescriptionType>` that belongs to `PartnerIdentification` namespace.

In this case, the algorithm 1 has been applied with RULE 2 and the result of `VERIFY@TYPE()` function refer to a `complexType`. For this, it has been transferred to an `ObjectProperty` by RULE 2.

5 CONCLUSIONS

In this paper, we have proposed an automatic transformation and mapping of the full RosettaNet B2B standard based on DTD or XML Schema to an ontological representation based on semantic web techniques using an OWL/XML rendering. Our goal is to make the transfer more transparent and reap the benefits of the effort put in the development of XML business standards and to mitigate the limits of previous approaches to the integration of semantics to B2B standards.

With semantic web technologies, industry acquires several benefits such as more efficient business interoperability and information exchange. As RosettaNet is a well organized B2B standards and its im-

plementations increase each year, it is important to study how to adapt it to semantic web techniques. For this, we proposed an approach for mapping RosettaNet PIPs to an ontological representation dealing with the conceptualisation used in XML Schema and DTD formats.

As future work, we suggest to use ontology matching and alignment approach to overcome the limits of our mapping.

ACKNOWLEDGEMENTS

We would like to thank RosettaNet Group for allowing the first author to develop his ideas by given us access to download the RosettaNet Partner Interface Processes (PIPs) available on the RosettaNet website (http://rosettanet.org/cms/sites/RosettaNet/).

REFERENCES

Battle, S. (2006). Gloze: XML to RDF and back again. In *Jena User Conference. May.: http://jena.hpl.hp.com/juc2006/proceedings. html.(Cit. on p.)*.

Cardoso, J. and Bussler, C. (2011). Mapping between heterogeneous XML and OWL transaction representations in B2B integration. *Data & Knowledge Engineering*, 70(12):1046–1069.

Dogac, A., Kabak, Y., and Laleci, G. B. (2004). Enriching ebXML registries with OWL ontologies for efficient service discovery. *Proceedings of the 14th International Workshop on Research Issues on Data Engineering: Web Services for E-Commerce and E-Government Applications (RIDE)*.

García, R. and Gil, R. (2007). Facilitating business interoperability from the semantic web. In *Business Information Systems*, pages 220–232. Springer.

Haller, A., Gontarczyk, J., and Kotinurmi, P. (2008). Towards a complete SCM ontology: the case of ontologising RosettaNet. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1467–1473. ACM.

Hitzler, P., Rudolph, S., and Krötzsch, M. (2009). *Foundations of semantic web technologies*. Chapman & Hall/ CRC.

Huemer, C. (2002). ebXML: An Emerging B2B Framework.

Kantola, P. and Korhonen, J. J. (2002). RosettaNet vs. ebXML–security solutions and exception handling. http://www.soberit.hut.fi/t-86/t-86.161/2002/rosettanet.

Kotinurmi, P., Haller, A., and Oren, E. (2011). *Global Business: Concepts, Methodologies, Tools and Application*, volume 4, chapter Ontologically enhanced RosettaNet B2B Integration, page 27. USA.

Lytras, M. and García, R. (2008). Semantic Web applications: a framework for industry and business exploitation–what is needed for the adoption of the semantic web from the market and industry. *International Journal of Knowledge and Learning*, 4(1):93–108.

Motik, B., Patel-Schneider, P. F., and Parsia., B. (2012). OWL 2 Web Ontology Language XML Serialization (Second Edition).

Panziera, L., Palmonari, M., Comerio, M., and De Paoli, F. (2010). WSML or OWL? a lesson learned by addressing NFP-based selection of semantic Web services. In *Non Functional Properties and SLA Management (NFPSLAM 2010) workshop.*

Robinson, P. and Bauer, S. (2011). *Introduction to bio-ontologies*, volume 41. Chapman & Hall.

Roman, D., Lausen, H., and Keller, U. (2005). Web Service Modeling Ontology standard. *WSMO Working Draft v1.0*.

RosettaNet Program Office (2011). *Overview Clusters, Segments, and PIPs*.

Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical OWL-DL reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2):51–53.

Tsinaraki, C. and Christodoulakis, S. (2007). XS2OWL: a formal model and a system for enabling XML schema applications to interoperate with OWL-DL domain knowledge and semantic web tools. *Digital Libraries: Research and Development*, pages 124–136.

W3C Member Submission, A. OWL-S: Semantic Markup for Web Services. http://www.w3.org/submission/ OWL-S/.

W3C Member Submission, A. Web Service Modeling Language (WSML). http://www.w3.org/submission/ WSML/.

Wang, J. and Song, Y. (2006). Architectures supporting rosettanet. In *Software Engineering Research, Management and Applications, 2006. Fourth International Conference on*, pages 31–39. IEEE.