Facilitating Business Interoperability from the Semantic Web

Roberto García, Rosa Gil

Universitat de Lleida
Jaume II 69, E-25001 Lleida, Spain
rgarcia@diei.udl.es
rgil@diei.udl.es

Abstract. Most approaches to B2B interoperability are based on language syntax standardisation, usually by XML Schemas. However, due to XML expressivity limitations, they are difficult to put into practice because language semantics are not available for computerised means. Therefore, there are many attempts to use formal semantics for B2B based on ontologies. However, this is a difficult jump as there is already a huge XML-based B2B framework and ontology-based approaches lack momentum. Our approach to solve this impasse is based on a direct and transparent transfer of existing XML Schemas and XML data to the semantic world. This process is based on a XML Schema to web ontology mapping combined with an XML data to semantic web data one. Once in the semantic space, it is easier to integrate different business standards using ontology alignment tools and to develop business information systems thanks to semantics-aware tools.

Keywords. Semantic Web, ontology, B2B, integration, mapping

1. Introduction

As more and more business is performed in the Internet and stronger inter-organisational links are established, enterprises needs for advanced information processing and data integration grow.

The objective is then to settle shared information spaces. The more mature initiatives come from standards bodies and enterprises organisations. They try to build common business model and languages and they usually root on standardised grammars based on XML technologies.

The previous approach finds difficulties due to the complexity of the business domain. Business languages grow more and more and it is difficult to manage them by computerised means if just their grammar is formalised. Moreover, different languages proliferate and their integration is almost impossible moving at just the syntactic level.

Therefore, recent approaches explore the possibilities of formal semantics through ontologies. They seem promising but, as they are recent, they lack momentum. Moreover, they find difficulties in getting it as they do not see support from the business world and stay as research issues.

Our approach to solve this impasse situation is to take profit from the great efforts that have been done in the XML e-business world and transfer them to the semantics-oriented one. More concretely, our objective is to map them to Semantic Web ontologies as they are a way of formalising semantics in a way that integrates smoothly in the Web.

The key point is that, as B2B standards are developed by people with domain concept models in their minds, they partially formalise their semantics while defining the XML Schemas for the different business languages. However, as XML Schemas are not semantics aware, this implicit semantics remain hidden from the computational point of view. They can be made explicit by mapping XML Schema constructs to the ontology language ones that correspond to their implicit semantics.

The previous XML Schema to Web Ontology mapping is combined with a transformation from XML data to RDF semantic data. Altogether, both mappings allow a transparent transfer of existing XML-based business data to the Semantic Web. Although there are other attempts to connect the XML and Semantic Web worlds, they just concentrate on the XML Schemas or the XML data so they do not achieve this level completeness and transparency in the transfer.

Once in the Semantic Web, it is possible to use semantics tools that make it easier to integrate data coming from disparate sources and to develop business management applications. For instance, it is possible to perform intelligent retrieval by semantic queries.

The rest of this paper is organised as follows. First, Section 2 presents the state of the art in the B2B domain. Then, in Section 3, the proposed methodology for a transparent transfer from XML-based B2B data to the Semantic Web is detailed. The results for this methodology when applied to some of the main B2B standards are shown in Section 4, together with the benefits obtained for data integration and information processing. Finally, Section 5 concludes the paper and presents the future work.

2. State of the Art

There are many B2B initiatives from international standardisation bodies and organisations and most of them are based on XML Schemas. For instance Biztalk[1], RosettaNet[2], ebXML[3] or UBL[4].

As it has been introduced, the XML approach does not scale well when the standardised domains are as complex as the business one. In this case, the lack of formal semantics makes it very difficult to automate sophisticated information processing mechanisms like integrating data from different standards and making them interoperate. Moreover, there is a lot of domain knowledge that remains hidden to implementations due to XML Schema expressive limitations. It is written down in the standards

[1] http://www.microsoft.com/biztalk

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

[3] Electronic Business using eXtensible Markup Language, http://www.ebxml.org

[4] Universal Business Language 1.0, http://docs.oasis-open.org/ubl/cd-UBL-1.0

specifications and must be read and interpreted each time an implementation of the standard is persuaded.

Consequently, many research efforts have moved to the formal semantics domain and there are many business modelling initiatives based on ontologies, and more concretely Semantic Web ontologies [1]. Most of them take profit from the enhanced expressive power so they try to build a complete enterprise model on the basis of a formal conceptualisation [2,3,4].

There are also other initiatives that are based on ontologies but that start from existing conceptualisations, which is an approach nearer to the one proposed in this paper. For instance, there is the eClassOWL ontology based on the products and services categorization standard eCl@ss [5], and EDI Ontology[5] that formalises EDI X12 Transaction Sets or the RosettaNet Ontology[6], an OWL implementation of RosettaNet Partner Interface Processes (PIPs).

However, these attempts to move B2B standards to the Semantic Web are based on an ad-hoc mapping from the source documents. This effort has to be done each time a new standard is mapped. Moreover, they just deal with the schema part so any existing data based on these standards is not semantically enriched. Therefore, what we get is a quite empty semantic framework. It has good conceptualisations but it lacks the semantic data it can operate on.

3. XML Semantics Reuse Methodology

On the contrary to the previous initiatives to B2B standards semantics formalisation, our approach provides an automatic and transparent mapping from XML Schema-based standard conceptualisations to OWL ontologies complemented with an XML B2B data to semantic RDF data one. The objective is to get a functional semantic framework full of semantic data.

Section 3.1 presents the related work and its limitations for the objectives stated in the introduction. These limitations have motivated the development of the XML Semantics Reuse Methodology. There is an overview of the proposed methodology in Section 3.2 and its two main components, the XML Schema to OWL and XML to RDF mappings, are detailed in sections 3.3 and 3.4. The architecture of the system that implements the methodology is presented in Section 3.5.

3.1. Related Work

This has been already detected as a key issue in order to add momentum to the Semantic Web so there are many attempts to move conceptualisations and data from the XML domain to the Semantic Web.

Some of them just model the XML tree using the RDF primitives [6] so there are not formalised semantics. The same happens when the approach is to encode XML

[5] http://www.wsmo.org/TR/d27/v0.1

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

semantics integrating RDF into XML documents [7,8] as the result is not semantic data, just ad-hoc XML.

In order to get semantic data, other initiatives concentrate on modelling the knowledge implicit in XML languages definitions, i.e. DTDs or the XML Schemas, using web ontology languages [9,10,11]. However, they do not generate transparent semantic data as the formalised semantics root on RDF ad-hoc semantics, i.e. they rely on custom RDF constructs that should be interpreted in a specific way in order to capture the formalised semantics. Therefore, existing applications that are not aware of these interpretations have no access to the semantics.

The most transparent approach we have found is Gloze XML to RDF [12]. It provides a direct mapping from XML data to RDF semantic data, which is based on the XML Schema implicit semantics. However, it does not fix the XML Schema semantics into an ontology. Therefore, it is not possible to work with the semantic data using semantic tools that rely on ontologies formal semantics or to perform semantic data integration at the conceptual level by mapping ontologies.

As it has been shown, none of them facilitates an extensive transfer of XML data to the Semantic Web in a general and transparent way. Their main problem is that the XML Schema implicit semantics are not made explicit when XML data instantiating this schemas is mapped.

Consequently, they do not take profit from the XML semantics and produce RDF data almost as semantics-blind as the original XML. Or, on the other hand, they capture these semantics but they use additional ad-hoc semantic constructs that produce less transparent data.

3.2. Overview

In order to overcome the limitations detected in the previous section, the "XML Semantic Reuse Methodology" [13] has been developed and implemented in the ReDeFer[7] project. It combines an XML Schema to OWL web ontology mapping, called XSD2OWL, with a mapping from XML to RDF, XML2RDF. The ontologies generated by XSD2OWL are used during the XML to RDF step in order to generate RDF data enriched by the XML Schema semantics made explicit.

3.3. XSD2OWL Mapping

The XML Schema to OWL mapping is responsible for capturing the schema implicit semantics. This semantics are determined by the combination of XML Schema constructs. The mapping is based on translating this constructs to the OWL ones that best capture their semantics. These translations are detailed in Table 1.

Some mappings depend on the context where the XML Schema construct appears. Therefore, XPath pointers have been used in order to detail this context, e.g. "complexType//element" refers to an element definition that appears inside a complex type definition.

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

Table 1. XSD2OWL mappings and shared semantics

XML Schema XPath	OWL	Shared informal semantics
element|attribute	rdf:Property owl:DatatypeProperty owl:ObjectProperty	Named relation between nodes or nodes and values
element@substitutionGroup	rdfs:subPropertyOf	Relation can appear in place of a more general one
element@type	rdfs:range	The relation range kind
complexType|group|attributeGroup	owl:Class	Relations and contextual restrictions package
complexType//element	owl:Restriction	Contextualised restriction of a relation
extension@base|restriction@base	rdfs:subClassOf	Package concretises the base package
@maxOccurs @minOccurs	owl:maxCardinality owl:minCardinality	Restrict the number of occurrences of a relation
sequence choice	owl:intersectionOf owl:unionOf	Combination of relations in a context

The XSD2OWL mapping is quite transparent and captures a great part XML Schema semantics. The same names used for XML constructs are used for OWL ones, although in the new namespace defined for the ontology. XSD and OWL constructs names are identical; this usually produces uppercase-named OWL properties because the corresponding element name is uppercase, although this is not the usual convention in OWL.

One of the key points is the *xsd:element* and *xsd:attribute* mapping. As it can be observed in Table 1, there are three alternatives, which are selected depending on the kind of values they can take as specified in the schema. All attributes are mapped to *owl:DatatypeProperty* as all of them have simple type values. This is also the case for elements that have a simple type value. On the other hand, if the value is a complex type, they must be mapped to an *owl:ObjectProperty*.

The third option is necessary as *xsd:elements* can have both simple and complex type values. In these cases, the element must be mapped to an *rdf:Property*, which can deal with both data type (for *xsd:simpleType*) and class instance (for *xsd:complexType*) values.

From the combination of all these XML Schema constructs mappings, XSD2OWL produces OWL ontologies that make the semantics of the corresponding XML Schemas explicit. The only caveats are the implicit order conveyed by *xsd:sequence* and the exclusivity of *xsd:choice*.

For the first problem, *owl:intersectionOf* does not retain its operands order, there is no clear solution that retains the great level of transparency that has been achieved. The use of RDF Lists might impose order but introduces ad-hoc constructs not present in the original data. Moreover, as it has been demonstrated in practise, the element ordering does not contribute much from a semantic point of view. For the second problem, *owl:unionOf* is an inclusive union, the solution is to use the disjointness OWL construct, *owl:disjointWith*, between all union operands in order to make it exclusive.

The XSD2OWL mapping has been checked using OWL validators, e.g. Pellet[8], which have been used in order to test the semantic consistency of the resulting ontologies.

3.4. XML2RDF Mapping

Once all the XML Schemas are available as OWL ontologies, it is time to map the XML data that instantiates them. The intention is to produce RDF data as transparently as possible. Therefore, a structure-mapping approach has been selected [6]. It is also possible to take a model-mapping approach [14], which is based on representing the XML information set using semantic tools. However, this approach reduces the transparency of the mapped data as it depends on the particular modelling decisions.

Transparency is achieved in structure-mapping models because they only try to represent the XML data structure, i.e. a tree, using RDF. The RDF model is based on the graph so it is easy to model a tree using it. Moreover, we do not need to worry about the semantics loose produced by structure-mapping. We have formalised the underlying semantics into the corresponding ontologies and we will attach them to RDF data using the instantiation relation *rdf:type*.

The structure-mapping is based on translating XML data instances to RDF ones that instantiate the corresponding constructs in OWL. The more basic translation is between relation instances, from *xsd:elements* and *xsd:attributes* to *rdf:Properties*. Concretely, *owl:ObjectProperties* for node to node relations and *owl:DatatypeProperties* for node to values relations.

However, in some cases, it would be necessary to use *rdf:Properties* for *xsd:elements* that have both data type and object type values. Values are kept during the translation as simple types and RDF blank nodes are introduced in the RDF model in order to serve as source and destination for properties. They will remain blank until they are enriched with type information from the corresponding ontology. The resulting RDF graph model contains all that can be obtained from the XML tree, as it is shown in Fig. 1.

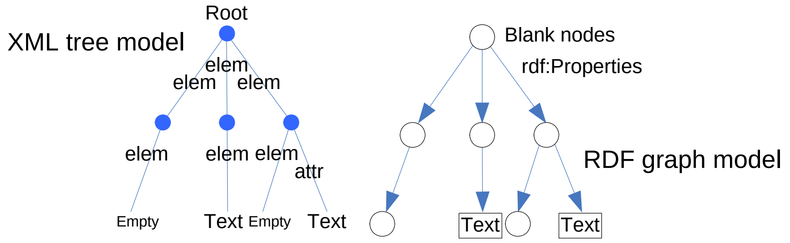

Fig. 1. RDF model for the XML tree

It is already semantically enriched thanks to the *rdf:type* relation that connects each RDF properties to the *owl:ObjectProperty* or *owl:DatatypeProperty* it instantiates. It can be enriched further if the blank nodes are related to the *owl:Class* that defines the package of properties and associated restrictions they contain, i.e. the corresponding

[8] Pellet OWL Consistency Checker, http://www.mindswap.org/2003/pellet/demo.shtml

xsd:complexType. This semantic decoration of the graph is formalised using *rdf:type* relations from blank nodes to the corresponding OWL classes.

At this point we have obtained a semantics-enabled representation of the input data. The instantiation relations can now be used to apply OWL semantics to data. Therefore, the semantics derived from further enrichments of the ontologies, e.g. integration links between different ontologies or semantic rules, are automatically propagated to instance data thanks to inference.

The XML2RDF mapping has been validated using some test XML instances. This test instances have been mapped to RDF and then back to XML. Then, it has been possible to compare the original and derived XML instances in order to detect mapping errors as, due to the mapping transparency, the underlying XML tree structure is preserved.

3.5. System architecture

Based on the previous XML world to Semantic Web domain mappings, a system architecture that facilitates B2B data integration and retrieval has been developed. The architecture is sketched in Fig. 2.

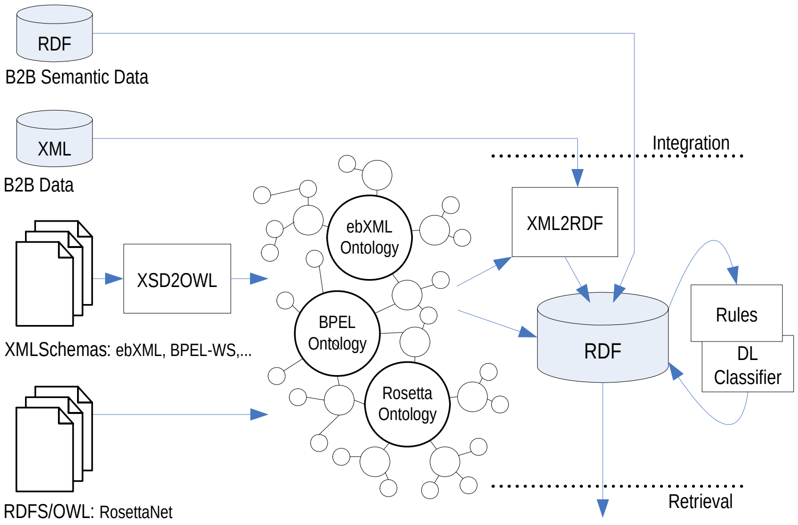

Fig. 2. B2B data integration and retrieval architecture

First of all, the architecture is fed with existing OWL ontologies, e.g. the RosettaNet Ontology, or those generated by XSD2OWL. For instance the ebXML or BPEL ontologies detailed in Section 4.1. These ontologies constitute the system semantic framework shown in the centre of Fig. 2.

Then, it is time to load B2B data. Semantic data can be directly fed into the system. On the other hand, B2B XML data must pass through the XML to RDF mapper. The mapping relies on the existing ontologies framework in order to produce semantically enriched RDF data.

All the data and ontologies are persisted in a RDF data store, where semantic tools like rule engines or semantic queries operate. Once all data has been put together, the semantic integration can take place. Finally, it is possible to perform intelligent data retrieval using semantic queries.

4. Preliminary Results

The XML Semantic Reuse Methodology has been tested with some XML-based B2B standards, ebXML and BPEL-WS. First, the XML Schemas for the standards selected parts have been mapped to OWL ontologies using the XSD2OWL mapping, as it is detailed in Section 4.1. Once the ontologies for these standards have been generated, automatic tools for ontology alignment have been used in order to derive some preliminary integration rules. This exercise shows how existing semantic web ontology alignment tools can be then applied in order to generate interoperability rules for B2B data based on these standards. Some of the integration rules are detailed in Section 4.2 while Section 4.3 shows the benefits of semantic queries compared to XML ones, fact exemplified with a piece of RDF mapped from XML B2B data.

4.1. Mapped Ontologies

Currently, the XML Semantics Reuse Methodology has been applied to some of the main B2B standards from the OASIS standardisation body. Their schemas are available online and they provide a good test bed for the XML Schema to OWL mapping and the subsequent semantic tools applications.

From the OASIS Electronic Business using eXtensible Markup Language (ebXML) initiative, the Business Process (ebBP) and Collaboration Protocol Profile and Agreement (ebCPPA) schemas have been considered. On the other hand, from the OASIS Web Services initiative, the Web Services Business Process Execution Language (BPEL-WS) schema has been selected.

This standards focus on business process modelling and on how collaborations among business parties are established in order to connect them. Although all of them come from the same standardisation body, there are not formal links among the corresponding schemas. Therefore, there are not integration means for them that facilitate interoperation.

All the resulting ontologies, together with some additional links and documentation, are available from the BizOntos web page[9]. Fig. 3 shows a portion of the ebBP class hierarchy. This hierarchy shows how the ontology makes the semantics in the XML extension relations among *xsd:ComplexTypes* explicit as a class hierarchy governed by inheritance relations.

[9] BizOntos, http://rhizomik.net/ontologies/bizontos

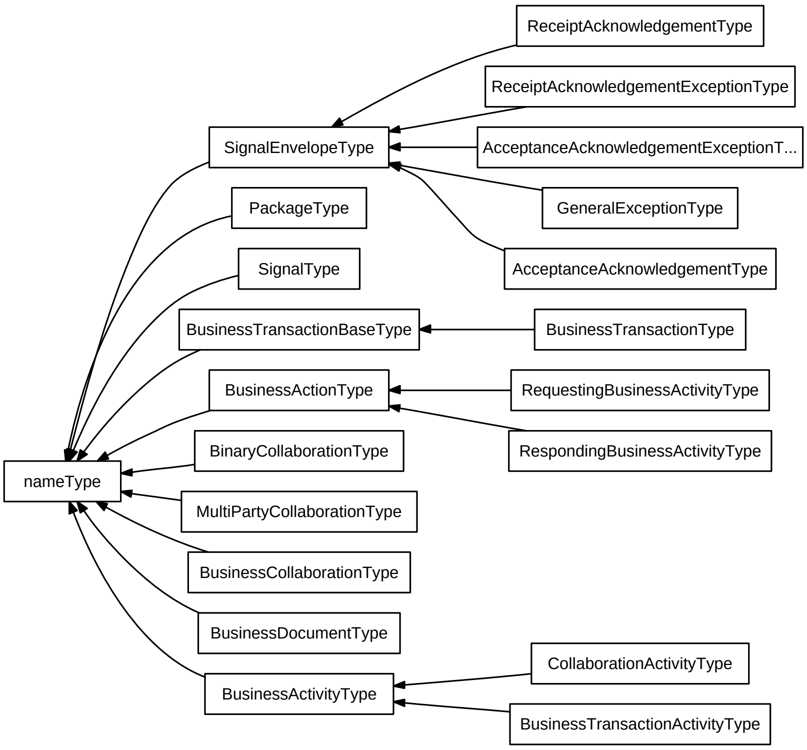
ReceiptAcknowledgementType
ReceiptAcknowledgementExceptionType
SignalEnvelopeType
AcceptanceAcknowledgementExceptionT...
PackageType
GeneralExceptionType
SignalType
AcceptanceAcknowledgementType
BusinessTransactionBaseType
BusinessTransactionType
BusinessActionType
RequestingBusinessActivityType
BinaryCollaborationType
RespondingBusinessActivityType
nameType
MultiPartyCollaborationType
BusinessCollaborationType
BusinessDocumentType
CollaborationActivityType
BusinessActivityType
BusinessTransactionActivityType

Fig. 3. Portion of the class hierarchy in the ebXML Business Process ontology

4.2. Semantic Integration

Once the semantics of the B2B standards are formalised as OWL ontologies, they are easily integrated using OWL semantic relations for equivalence and inclusion: *subClassOf*, *subPropertyOf*, *equivalentClass*, *equivalentProperty*, *sameIndividualAs*, etc.

These relationships capture the semantics of the data integration. Moreover, this integration can be partially automated using ontology matching methods [15]. Some of these mappings generated by the OWL Ontology Aligner[10] are presented in the following subsections 4.2.1 and 4.2.2, which are devoted to the ebBP to ebCPPA mapping and the ebBP to BPEL-WS mappings respectively.

With all the relevant integration mappings, semantically-enriched B2B data coming from different standards is automatically integrated. The OWL relations formal semantic allow inference engines to derive the necessary mappings among instance data.

[10] OWL Ontology Aligner, http://align.deri.org

4.2.1. ebXML BP to ebXML CPPA mappings

Some of the mappings derived by the OWL Ontology Aligner for the ebXML Business Process (ebBP) ontology and the ebXML Collaboration Protocol Profile and Agreement (CPPA) ontology are shown in Table 2. These are equivalence and inclusion relations for related ontology classes and properties.

Table 2. ebXML BP to ebXML CPA mappings

ebbp:location	≡	cppa:location
ebbp:Role	≡	cppa:Role
ebbp:type	≡	cppa:type
ebbp:mimeType	≡	cppa:mimetype
ebbp:RoleType	≡	cppa:RoleType
ebbp:ProcessSpecificationType	≡	cppa:ProcessSpecificationType
ebbp:Start	≡	cppa:Start
ebbp:ProcessSpecification	≡	cppa:ProcessSpecification
ebbp:value	≡	cppa:value
ebbp:name	≡	cppa:name
ebbp:uri	≡	cppa:uri
ebbp:BusinessAction	⊆	cppa:action
ebbp:isAuthenticated	≡	cppa:authenticated
Specification	⊆	cppa:ProcessSpecification
...		...

4.2.2. ebXML BP to BPEL-WS

Some mappings for the ebXML Business Process (ebBP) ontology to the Business Process Execution Language for Web Services (BPEL-WS) ontology are shown in Table 3. They have been also generated by the OWL Ontology Aligner.

Table 3. ebXML BP to BPEL-WS mappings

ebbp:expression	≡	bpel:expression
ebbp:expressionLanguage	≡	bpel:expressionLanguage
ebbp:name	≡	bpel:name
ebbp:pattern	≡	bpel:pattern
ebbp:Variable	≡	bpel:variable
ebbp:PreCondition	⊆	bpel:condition
ebbp:BusinessPartnerRole	≡	bpel:partnerRole
...		...

4.3. Semantic Applications

Once the B2B ontologies and the integration rules have been generated, it is time for the semantic enrichment of B2B data corresponding to the mapped standards. For instance, Table 4 shows a portion of the RDF data generated by the XML2RDF mapping from a CPPA XML example. As it can be observed, the input XML data tree is modelled using RDF properties and it is enriched with typed blank nodes. These types correspond to the XML Schema complex types mapped to OWL classes during the XSD2OWL step.

Table 4. Part of the RDF version of cpp-example.xml

```
<rdf:RDF xmlns:tp="...ontologies/2006/06/cpp-cpa-v1_0.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<tp:CollaborationRoleType rdf:about=".../cpp-example.rdf#N00">
    <tp:ProcessSpecification>
      <tp:ProcessSpecificationType>
        <tp:name>buySell</tp:name>
        <tp:version>1.0</tp:version>
      </tp:ProcessSpecificationType>
    </tp:ProcessSpecification>
    <tp:Role>
      <tp:RoleType>
        <tp:name>buyer</tp:name>
      </tp:RoleType>
    </tp:Role>
    <tp:CertificateRef>
      <tp:CertificateRefType>
        <tp:certId>N03</tp:certId>
      </tp:CertificateRefType>
    </tp:CertificateRef>
    ...
  </tp:CollaborationRoleType>
  ...
</rdf:RDF>
```

Thanks to this semantic enrichment, it is possible to take profit from the XML Schema implicit semantics using Semantic Web tools. Therefore, it is not even required to add more semantics to the resulting ontology; just with the semantics implicit in the original XML Schema, it is possible to provide new functionalities. For instance, the classes' hierarchy, which is formalised in the ontology and linked to the RDF data through the typed blank nodes, can be used by semantic query engines.

In order to illustrate this, a hypothetical ebBP application scenario can be considered. One of its functionalities is to process all the nodes referring to signal envelope types, which correspond to the *SignalEnvelopeType* complex type. If an XML-based tool like XQuery is used to do that, we must be aware of the implicit hierarchy of segment types and implement an XQuery that considers all possibilities, i.e. *ReceiptAcknowledgementType*, *GeneralExceptionType*, *AcceptanceAcknowledgementType*, etc.

On the other hand, once the hierarchy of segments types is available in OWL form, semantic queries benefit from the now explicit semantics. Therefore, a semantic query for *SignalEnvelopeType* will retrieve all subclasses without requiring additional developing efforts. Table 5 shows a query based on the SPARQL semantic query language [16] which demonstrates the simplicity of a query capable of dealing with all the kinds of *SignalEnvelopeType*.

This is so because, although XML Schemas capture some semantics of the domain they model, XML tools are based on syntax. The captured semantics remain implicit from XML processing tools point of view. Therefore, when an XQuery searches for a *SignalEnvelopeType*, the XQuery processor has no way to know that there are many other kinds of segment types that can appear in its place, i.e. they are more concrete kinds of segments.

Table 5. SPARQL semantic query for *SignalEnvelopeType*

```
PREFIX ebbp: <http://rhizomik.net/ontologies/2006/06/ebbp-2.0.3.owl#>
SELECT ?r
WHERE (?r <rdf:type> ebbp:SignalEnvelopeType)
```

Additionally, semantic queries use integration rules, like the ones presented in Section 4.2, in order to perform B2B data integration. For instance, when a semantic query for "ebbp:location" elements is performed, the "ebbp:location ≡ bpel:location" is interpreted and due to the semantics it carries the query also retrieves "bpel:location" elements automatically. Therefore, ebPB and BPEL-WS data is integrated and processed seamlessly by existing ebPB or BPEL-WS semantic applications.

5. Conclusions and future work

As it has been shown, the increase of B2B interactions and their complexity make existing standards, most of them based on XML technologies, very difficult to manage and to make them interoperate. Consequently, there are many initiatives trying to profit from ontologies greater level of expressivity and the formal semantics they provide in order to make B2B management and interoperability easier.

However, these initiative lack momentum as most of the available B2B systems and data are based on XML tools. In order to mitigate this, this paper proposes to apply the XML Semantics Reuse Methodology, which contributes a complete and transparent transfer of B2B data from existing XML-based standards to the Semantic Web. It is based on a generic XML Schema to OWL mapping and complemented with an XML data to RDF mapping.

As it has been shown, XSD2OWL is used to map some of the main B2B standards XML Schemas and generate the corresponding OWL ontologies. These ontologies provide the anchor points for semantic integration and ontology matching tools are used to align them and derive integration rules.

These rules are used to integrate XML B2B data once it is mapped to RDF and semantically-enriched by the XML2RDF mapping. Then, semantic tools can be used to facilitate B2B applications development, e.g. semantic query engines. Semantic queries take profit from the complex types and elements hierarchies implicit in the XML Schema definitions.

All this process is transparent as it is based on a structure-mapping approach that preserves the original XML tree structure. Therefore, it is easy to go back to XML data if just the RDF graph properties are considered, which correspond to the XML data elements.

The future plans concentrate now on performing detailed alignments among the generated OWL ontologies in order to get a complete data integration framework. Another objective is to integrate the ontologies mapped from XML Schemas with existing ontologies like the Enterprise Ontology. These are rich ontologies with much more formal semantics as they are developed from the beginning using more expressive tools. If they are aligned with XSD2OWL ontologies, it will be possible to reuse the detailed semantics captured by this ontology in order to implement advanced B2B applications operating on existing XML-based standards.

References

1. Singh, R; Iyer, L.S.; Salam, A.F. (ed.): The Semantic e-Business Vision. Communications of the ACM, 48(12):38-77
2. Albani, A.; Dietz, J. L. G.; Zaha, J. M.: Identifying Business Components on the basis of an Enterprise Ontology. In: Konstantas, D.; Bourrières, J.-P.; Léonard, M.; Boudjlida, N. (ed.): Interoperability of Enterprise Software and Applications. Springer, pp. 335-347, 2006
3. Uschold, M.: King, M.; Moralee, S.; Zorgios, Y.: The Enterprise Ontology. The Knowledge Engineering Review, 13(1):31-89, Cambridge University Press, 1998
4. Dietz, J. L. G.: Enterprise Ontology. In: Chen, C.-S.; Filipe, J.; Seruca, I.; Cordeiro, J. (ed.): ICEIS 2005, Proceedings of the Seventh International Conference on Enterprise Information Systems, 2005
5. Hepp, M.: Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards. Journal on Semantic Web and Information Systems, 2(1):72-99, 2006
6. Klein, M.C.A.: Interpreting XML Documents via an RDF Schema Ontology. ECAI Workshop on Semantic Authoring, Annotation & Knowledge Markup. Lyon, France, 2002
7. Lakshmanan, L.; Sadri, F.: Interoperability on XML Data. Lecture Notes in Computer Science Vol. 2870, pp. 146-163. Springer, 2003
8. Patel-Schneider, P.F.; Simeon, J.; The Yin/Yang web: XML syntax and RDF semantics. Proceedings of the 11th World Wide Web Conference, pp. 443-453, Hawaii, USA, 2002
9. Amann, B.; Beeri, C.; Fundulaki, I.; Scholl, M.: Querying XML Sources Using an Ontology-Based Mediator. In R. Meersman; Z. Tari (ed.): On the Move to Meaningful Internet Systems 2002, ODBASE'02. LNCS Vol. 2519, pp. 429-448, Springer, 2002
10. Cruz, I.; Xiao, H.; Hsu, F.: An Ontology-based Framework for XML Semantic Integration. 8th Database Engineering and Applications Symposium, Portugal, 2004
11. Halevy, A.Y.; Ives, Z.G.; Mork, P.; Tatarinov, I.: Piazza: Data Management Infrastructure for Semantic Web Applications. Proceedings of the 12th World Wide Web Conference, pp. 556-567, Budapest, Hungary, 2003
12. Battle, S.: Gloze: XML to RDF and back again. In Proceedings of the First Jena User Conference, Bristol, UK, 2006
13. García, R.: Chapter 7: XML Semantics Reuse. In: A Semantic Web Approach to Digital Rights Management. PhD Thesis, Technologies Department, Universitat Pompeu Fabra, Barcelona, Spain, 2006. Available at http://rhizomik.net/~roberto/thesis/html/Methodology.html#XMLSemanticsReuse
14. Tous, R.; García, R.; Rodríguez, E.; Delgado, J.: Architecture of a Semantic XPath Processor. In: E-Commerce and Web Technologies, 6th International Conference. Springer, Lecture Notes in Computer Science Vol. 3590, pp. 1-10, 2005
15. Noy, N.: Semantic Integration: A Survey of Ontology-based Approaches. Sigmod Record, Special Issue on Semantic Integration, 2004
16. Prud'hommeaux, E.; Seaborne, A.: SPARQL Query Language for RDF. W3C Candidate Recommendation 6 April 2006. Retrieved July 4, 2006, from http://www.w3.org/TR/rdf-sparql-query