Extracting OWL Ontology from XML instances via XML Schema

Pham Thi Thu Thuy, Young-Koo Lee and SungYoung Lee
Dept. of Computer Engineering, KyungHee University, Yongin, Korea
{tttpham, sylee}@oslab.khu.ac.kr, yklee@khu.ac.kr

Summary

Currently, XML and its schema language have become the standard for data representation and information exchange format on the current web. Unfortunately, problems happen when integrating different data sources since XML mainly supports the document structure but lack consideration on sharing knowledge of data. Meanwhile, Semantic Web technologies, such as Web Ontology Language (OWL), can include the structure as well as the semantics of the data . Therefore, finding a way to integrate XML data as OWL ontology receives a high interest nowadays. In this paper we present a mapping notation to convert XML Schema to OWL domain knowledge and an effective method to transform XML instances into OWL individuals. While keeping the XML original structure, our work also adds more semantics for the XML document. Moreover, whole of the transformation processes are done automatically without any user interference. Further, our transforming approach provides the solution for duplicate element names in XML document which has not mentioned in the previous work. Our results in existing OWL syntaxes can be loaded immediately by OWL editors and Semantic Web applications.

1. Introduction

XML specifies the document structure and nowadays it becomes the standard format for storing and exchanging data on the current web [1]. Users can predefine the structure of XML documents by writing a DTD (Document Type Definition) or an XML Schema. Because XML Schema is an XML-based language and it supports data types and namespaces [2], many developers nowadays use XML Schema, instead of DTD, to create an XML document instead of DTD. Therefore, XML Schema is usually used as a standard mechanism to interchange information on the web. The main success of XML is its flexibility. It allows anyone to describe any content easily by creating their own tags. However, this freedom can cause lack of understanding between a document's author and its consumer. Since an object can be expressed by different vocabularies, it is hard for computer to recognize and differentiate the meaning of data. For instance, "author" can be described as "creator", "inventor", etc. according to users' opinions. Furthermore, XML exposes disadvantages when coming to the semantic interoperability. XML mainly focuses on the grammar but there is no way to describe the semantics of a document [3]. Therefore, problem happens when software agents would like to understand and reason about these XML data.

To solve this problem, many approaches have been proposed to narrow the gap between XML and ontology. Typical approaches on transforming XML Schema to OWL ontology are [4, 5, 6]. Their transformations are developed in XSLT by automatically mapping each definition in XML Schema to corresponding OWL domain knowledge. However, they only describe the mapping notations from XML Schema to OWL ontology and do not execute the transforming from XML instances into OWL individuals. Also focusing on OWL target, but authors of [7, 8, 9, 10, 11] intend to define a set of mapping rules from each schema to OWL ontology. These methods add more semantics for existing XML Schema but these rules are too complicated. They are produced by authors. Therefore, this set of rules may be different to each other even though they describe for a same schema. Moreover, it is impractical to define a set of mapping rules for each schema available on the internet, especially when it has large size.

Generally, existing schema mappings and XML transforming solutions still expose several limitations. Most of them try to narrow the gap between the XML Schema and OWL but do not solve the problems when same elements in an XML document appear. In this paper, beside a mapping step, we provide a method to transform XML documents into existing OWL ontology automatically and OWL result can be loaded immediately by Semantic Web applications.

2. XML Schema Mapping and XML Transforming

2.1 XML Schema mapping

In this stage, we create the collection of classes and properties from the given XML Schema as an input. This collection will be used to model data in the next step. The mapping notation from XML Schema to RDF Schema is shown in Table 1.

TABLE 1: XML SCHEMA MAPPING

XML Schema concepts	OWL concepts
Complex-type element	owl:Class,owl:ObjectProperty
Simple-type element	Property
Attribute	Property
Type	Datatype
minOccurs, maxOccurs	minCardinality, maxCardinality
sequence, all	intersectionOf
choice	Combination of "intersectionOf" and "complementOf"

In order to describe nesting classes, we add new definition "has_child_name", *child_name* is the name of that child element. Moreover, because XML allows same elements appeared within a document but OWL does not, when generating OWL individuals from XML instances, if the current XML element has the same name with the previous element, our procedure renames it by adding parent node's name concatenating with the symbol "_" before this element's name.

2.2 XML transforming

The input of this step is the OWL model generated from previous step and the given XML instance. Our illustrated example “moviedb.xml” contains the information about *movie* database. There are 41 movies in this file, each of them includes information about *title, actors, directors, year, etc.*. A part of that sample file is as below:

```
<?xml version="1.0"?>
<movielist>
  <movie id="1">
   <title>21 Grams</title>
    <audiochannels>
     <language>EN</language>
    </audiochannels>
   <subtitles> </subtitles>
   <country>USA</country>
   <year>2003</year>
   <genres> <genre>Drama</genre>
           <genre>Thriller</genre> </genres>
   <actors>
      <actor>Naomi Watts</actor>
     <actor>Benicio Del Toro</actor> </actors>
   <runtime in="min">125</runtime>
    <director>Alejandro
                  Gonzalez Inarritu</director>
   …   </movie>
</movielist>
```

And its corresponding XML schema as following:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="movielist">
   <xs:complexType>
   <xs:sequence>
   <xs:element maxOccurs="unbounded"
                 name="movie">
<xs:complexType mixed="true">
       <xs:sequence>
     <xs:element name="title" type="xs:string" />
     <xs:element name="plot" type="xs:string" />
            <xs:element name="audiochannels">
             <xs:complexType> <xs:sequence>
            <xs:element name="actors">
             <xs:complexType>
               <xs:sequence>
                <xs:element maxOccurs="unbounded"
                name="actor" type="xs:string" />
               </xs:sequence>
             </xs:complexType>
            </xs:element>  ...
```

3. Conclusion

The DTD2OWL framework presented in this paper allows the automatic mapping DTD to OWL domain knowledge and transforming XML instances into OWL individuals. Our procedure outperforms the existing methods due to the following five reasons. Firstly, while transforming all the elements of an XML document into OWL, our algorithm retains the original structure and captures the implicit semantics expressed in the XML document. Secondly, components in DTD are considered as classes or properties or data types based on their definitions and detail descriptions, this makes the result be independent from users' opinions. Thirdly, languages used in our procedure do their jobs as their original functions. DTD is used for defining XML structure, XML for describing data, OWL for providing definitions and relationships between data. Fourthly, our approach provides new method for transforming XML instances into existing OWL individuals without any user intervention. That method makes many XML documents to be converted to the OWL formats. Finally, during the transformation process, our procedure not only solves the duplicate element problems but also provides the inheritance mechanisms which help reducing the consumed memory. We hope that our research has created a bridge to narrow the gap between the XML data and OWL ontology. If this procedure is executed, a large amount of the XML data on the current Web will be interpreted into OWL ontology which is useful for the Semantic Web applications.

Further improvement to our work may be focused on the transforming XML Schema into OWL ontology by giving more semantics than current approaches. Moreover, in order to prove the quality of our transformation, we are going to extend our work to the semantics measurement. The structure and semantic similarity of XML and OWL documents will be computed and compared to other methods.

Acknowledgments

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2009-(C1090-0902-0002)). Also, it was supported by the IT R&D program of MKE/KEIT, [10032105, Development of Realistic Multiverse Game Engine Technology].

References

[1] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, F. Yergeau: Extensible Markup Language 1.0 (Fifth Edition). W3C, http://www.w3.org/TR/REC-xml/ (2008)

[2] XML Schema Language Comparison, Wikipedia, available at: http://en.wikipedia.org/wiki/XML_Schema_Language _Comparison (2009)

[3] Hawke, S.: XML with Relational Semantics: Bridging the Gap to RDF and the Semantic Web. W3C, http://www.w3.org/2001/05/xmlrs (2001)

[4] H. Bohring, S¨oren Auer: Mapping XML to OWL Ontologies. Marktplatz Internet: Von e-Learning bis e-Payment, Leipziger Informatik-Tage (LIT2005), pp. 147-156, Germany (2005)

[5] C. Tsinaraki, S. Christodoulakis: XS2OWL: A Formal Model and a System for Enabling XML Schema Applications to Interoperate with OWL-DL Domain Knowledge and Semantic Web Tools, DELOS Conference’2007, pp. 137-146 (2007)

[6] Chrisa Tsinaraki and Stavros Christodoulakis, “Interoperability of XML Schema Applications with OWL Domain Knowledge and Semantic Web Tools”, pp. 850–869, OTM Conference, Springer-Verlag , 2007.

[7] B. Amann, C. Beeri, I. Fundulaki, M. Scholl: Ontology-Based Integration of XML Web Resources, First Int. Semantic Web Conference, pp. 117-131, Springer-Verlag (2002)

[8] Toni Rodrigues, Pedro Rosa, Jorge Cardoso: Mapping XML to Existing OWL Ontologies. International Conference WWW/Internet 2006, pp. 72-77 (2006)

[9] C. Cruz, C. Nicolle: Ontology Enrichment and Automatic Population from XML Data, the 4th International VLDB Workshop on Ontology-based Techniques for Databases in Information Systems and Knowledge Systems, ODBIS 2008.

[10] Nassim KOBEISSY, Marc GIROD GENET and Djamal ZEGHLACHE, “Mapping XML to OWL for seamless information retrieval in context-aware environments”, pp.349-354, IEEE International Conf. on Pervasive Services, 2007.

[11] Philipp Kunfermann, and Christian Drumm, ”Lifting XML Schemas to Ontologies – The concept finder algorithm”, Mediate 2005 workshop.